Project Proposal | RF Scanner

Andre Magill, John Docalovich, Joey Pye, Zack Bennett, Shane Ryan
Advised by Dr. Jonathan Chisum

Introduction

Our project will attempt to scale down the existing RadioHound hardware system while also making it more power efficient and less expensive. Currently, the hardware is large, bulky and costly; we will create a smaller system integrated onto a PCB that is more cost efficient and provides a faster method of scanning than the present system. Through this, we wish to create large numbers of these low cost devices to increase the coverage area we can monitor while significantly cutting the overall price.

Problem Description

The RF scanning hardware currently being used by Dr. Chisum's group is known as RadioHound. In the current design, the RadioHound components include a Raspberry Pi, a custom PCB MANET, and RTL-SDR USB dongle. The Raspberry Pi acts as a relay; receiving commands from the Cloud and sending them to an MSP430 microcontroller located on the custom PCB. The RTL-SDR is a software defined radio that scans any 2 MHz band within the range of 25 MHz to 6 GHz. These signals are converted to a lower IF frequency for digitizing with a local oscillator and mixer by the MSP430. The Raspberry Pi then takes this captured data, sends it to the Cloud, and the data is plotted on a heatmap on a web interface.

The problem faced in this project involves taking the current designs of the RadioHound and redesigning a low-cost, low-power version of these distributed spectrum sensors; thus making it possible for applications in electronic warfare including locating enemy radar, IED triggers, and any technologies used over the RF spectrum. Another application of a remodeled RadioHound would include wireless communications coverage analysis and spectrum sharing for cellular companies or the FCC.

One of the main issues with the current design is that it consumes large amounts of power. For example, the Raspberry Pi, custom PCB, and the RTL-SDR dongle, together consume approximately 8 Watts of power. This requires a large battery during field-operation, and it would also limits its use to a day. Professor Chisum has given us a power budget of only 2.5W based on the components that are available to us now. The other possible optimization value would be the price of the RadioHound. With the Raspberry Pi 3 typically going for about $60 and the RTL-SDR being approximately $25, there is much improvement needed to be able to deploy the RadioHound in massive quantities for real world applications. Without this, there is a limited number of systems that can be put in an area, which creates low-resolution heatmaps.

Not only is power consumption and price an issue, the entire system is very slow. The FFT, Raspberry Pi, USB dongle, and custom PCB together process and record the data at a remarkably slow rate. This slows down the entire system, making data analysis and representation a tedious process.

Proposed Solution

One of the most significant changes that we will be making to the current RadioHound will be swapping out the MSP430 for an MSP432. There are many reasons for doing this including the fact that it will decrease power consumption, lower price, and simplify the process of device communication. As far as the power consumption, the MSP432 will replace the functions of the Raspberry Pi and RTL-SDR. With included ADC capabilities, power is not wasted on either of these replaced components. Not only this, but the MSP432 uses about two-thirds of the power of the MSP430. As far as pricing optimization, getting rid of two components altogether clearly makes the system less expensive by about $85; additionally, upgrading to a MSP432 only costs about $3 extra.

The SimpleLink host MCU wireless application microcontroller was created by the same company, and specifically built for the MSP432. The SimpleLink comes with communication stacks that make programming the interactions and data transfers with the MSP432 significantly easier to program than with an alternate WiFi chip. Using both this microcontroller and the corresponding wireless module also minimizes possible errors or miscommunications during setup and use.

A GPS component will be added to the system using an Adafruit MTK3339 priced at about $30. This will provide the RadioHound with the ability to locate where each device is located. The Adafruit GPS also provides a time signal so that the system is able to synchronize all of the sensors, eliminating the task of having to figure out when and where each piece of data is coming from during analysis. Another feature of the GPS component is that it is extremely small (16mm x 16mm x 5mm) and light (4 grams).

The power supply will most likely be a LiPo battery on the chip that will be able to run for 24 hours at a 1% duty cycle. Depending on the current draw and power capabilities, we will tailor the battery choice to be the smallest, lightest, and least expensive possible while still meeting requirements. This will allow the battery to actually fit in the RadioHound housing (which it currently does not), as well as making it more efficient than the massive battery that is currently being used. We are also currently considering both rechargeable and disposable depending on who our potential customer might be for this technology.

An analog circuit will be added to convert the signal from the frequency domain to a DC voltage signal that can be sampled by the microcontroller. This circuit, which will most likely be an arrangement of resistors, diodes, and capacitors, will be able to integrate the incident power in the frequency bin without doing an FFT. This will speed up the process of data analysis since the analog analysis is much faster than the digital.

Demonstrated Features

With a design as complex as this, there are certain features that we will need to demonstrate to show that our project has been a success. First and foremost, we must be able to successfully integrate the entire system onto a single PCB. Moreover, we need to be able to create a device that draws less than the 2.5W of power, as allotted to us by Professor Chisum. This will allow us to power the system off a battery cell possibly with recharge capabilities using solar panels. We therefore must be able to change the power source during operation: battery with solar

cells or wall power. The total cost of each individual device should also be less than $100 to be economically appealing and allow for large numbers of these sensors to be deployed. Finally, the smaller chips should be able to capture at a rate equal to or faster than the current system. This may be achieved while the boards operate at a low duty cycle to conserve power.

Available Technologies

1. GPS Module MTK3339 chipset
    a. -165 dBm sensitivity, 10 Hz updates, 66 channels
    b. Ultra low power usage: 20mA current draw while tracking
    c. 3.3V operation
    d. Built-in data logging
    e. PPS output on fix
    f. works up to ~32 Km altitude
    g. Ultra small size: only 16mm x 16mm x 5mm and 4 grams
2. MSP432P401R (TI microcontroller)
    a. 48 MHz frequency
    b. 64 kB of RAM
    c. 256 kB Flash Main Memory
    d. 80 uA/MHz active power
    e. 16 Bit Precision ADC converters with 24 channels
    f. 1 MSPS Rate
3. CC3220 (single chip wireless MCU)
    a. SimpleLink WiFi and IoT Wireless MCU
    b. Optimized Low Power capability
    c. SPI/UART/I2C/ADCs/27 GPIOs
    d. WiFi TX Output Power 18.0 dBm/WiFi RX Sensitivity -96 dBm

Engineering Content

1. GPS module - Design a system with the current GPS chip that will provide a consistent time signal between the devices as well as an accurate location for each of the devices. Following the build criteria for the chip will allow us to implement them into the circuit. A test of the system will be if the position is accurately recorded for the device and the devices sync to one another based on the time signature from these modules.

2. Power module - Design a smart power system that can decide which available power source can/should be used to power the device. Potentially, we will implement a rechargeable battery with a small solar panel and a steady outside source (wall power). A test for the power module is if it can successfully switch from one power source to another when more than one is present but one is not sufficient or is not desirable: e.g. not using the battery

when wall power is present.

3. Analog circuit - Design a circuit that takes in the analog input signal and outputs a DC voltage proportional to the input signal. This signal is then fed to the microcontroller. The challenge of the circuit is integrating power into the signal without performing an FFT as this greatly slows the rate of the device. A test would be to do a frequency spectrum analysis with an oscilloscope and then calibrate the various DC voltage levels to the corresponding power levels.

4. Microcontroller - Programmed to fulfill its various functions. It programs the phase locked loop variable local oscillator and the variable gain amplifier to scan the various frequency bins and amplify the IF signal. It also processes the proportional DC voltage, digitizing it and sending it to the WiFi chip to be sent out on the network. Moreover, it communicates with the GPS chip to timestamp the data and synchronize with the other sensors. It also controls the battery and the duty cycle: only running at 1% duty cycle when dependent on the onboard battery or running at 100% when the large battery is plugged into the sensor. Finally, it contains a cyclic memory buffer to continually store the DC voltage data when no WiFi network is available.

5. Wireless - Design a wireless system that is controlled by the microcontroller and allows the data collected to be sent out for further processing and to be viewed by the user. The challenge of this is to construct a method where the devices can "piggyback" off of Dr. Chisum's current MANET setup or act individually without the MANET board. One method we are considering is a cyclic writing cycle to act as a memory buffer where the most recent data is always being recorded. The test for this system is if the devices can communicate with one another to transport the data to the server as well as accurately relay the data back to the user.

Conclusions

This project aims to solve the problem of real-time, cheap frequency spectrum mapping. Our design will allow for a low cost, faster version of the current RadioHound hardware that is completely integrated onto one PCB. While still dependent on the full version being designed by Dr. Chisum's group, our design will fit into a sensor hierarchy to allow the full version to quickly zero in on the specific frequency range that is being broadcasted on. Moreover, since this design aims to be significantly cheaper, more of these sensors can be deployed in order to provide more accurate frequency maps.

Our next steps are acquiring the MSP432 development boards so that we can begin to integrate Dr. Chisum's group's code with our own. We also will assign the various functional blocks to team members. Moreover, we will begin working with the SimpleLink chip to configure the WiFi protocols for our microcontroller.