AN ANDROID SMART PHONE INTERFACE FOR A DYNAMIC DIRECTION OF ARRIVAL SYSTEM

A Senior Scholars Thesis

by

DANIEL LEE REVIER

Submitted to Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

May 2012

Major: Electrical Engineering

AN ANDROID SMART PHONE INTERFACE FOR A DYNAMIC DIRECTION OF ARRIVAL SYSTEM

A Senior Scholars Thesis

by

DANIEL LEE REVIER

Submitted Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as

UNDERGRADUATE RESEARCH SCHOLAR

Approved by:

Research Advisor: Gregory Huff
Director for Honors and Undergraduate Research: Duncan McKenzie

May 2012

Major: Electrical Engineering

ABSTRACT

An Android Smart Phone Interface for a Dynamic Direction of Arrival System.
(May 2012)

Daniel Lee Revier
Department of Electrical and Computer Engineering
Texas A&M University

Research Advisor: Dr. Gregory Huff
Department of Electrical and Computer Engineering

The Global Positioning System (GPS) remains well suited for mobile applications in open areas where the reception of beacon signals from satellites remains in (or near) a line-of-sight path; however, positioning inaccuracies and other operational difficulties are common once in the interior of buildings and urban canyons of large cities where coverage is obstructed. As a result, this work focused on the investigation and development for a new method of multiple target tracking in comparatively smaller areas to complement the shortcomings of GPS. These small networks were implemented via Android OS mobile phone devices in coordination with a dedicated direction-of-arrival (DOA) server through a polarization reconfigurable antenna. The phones were able to connect to the server, access an array of DOA information for themselves and other users, and then display the information in a practical manner on the screen. A Bluetooth communication link was used in order to improve the feasibility of the application. The advantages of this methodology over traditional GPS include near real-time tracking of targets and a higher location resolution, which has the potential to aid applications varying from military to commercial in nature.

DEDICATION

To my parents and God, without either I would not be as successful as I have been.

ACKNOWLEDGMENTS

Caterpillar, Inc. for their generous support through the Caterpillar Research Scholarship.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

Smart phones, tablets, and other portable electronic devices provide considerable access to the GPS for the layperson. A growing number of navigation and geolocation applications (or apps) for public, private, and government users make use of this unique capability to enable tracking, tagging, or monitoring features and improve the augmented reality experience. Connectivity to the Global Navigation Satellite System (GNSS) drives this proliferation of localization capabilities but accuracy and access to these features can be severely limited or even unobtainable inside warehouses and large buildings, urban canyons, underground facilities, and other areas where physical limitations imposed by shadowing, attenuation, and other local factors contribute to an electromagnetically harsh environment. Currently, the methods developed in [1] - [4] reshape the way position is derived in these environments; however, it is believed that a greater degree of accuracy and certainty could be achieved with a dynamic direction-of-arrival (DOA) tracking system specifically tailored for these domains.

Local Positioning Systems (LPS) provide a means to obtain localization in areas of limited GPS coverage or blind spots by utilizing preexisting wireless infrastructural and/or stand-alone systems using wireless waypoints. Referencing the geographical location of a nearby Wi-Fi router, triangulating using multiple cell towers, and pinpointing various Radio-Frequency Identification (RFID) tags are examples of LPS. There are considerably more techniques that can be used to accomplish this, but in nearly all implementations of LPS the relative position of a mobile device can be derived simply by wireless communi-

---

This thesis follows the style and format of *IEEE.*

cation. This often provides a very cost effective solution but many methods cannot offer highly accurate and near-real-time position estimations of a mobile device or RFID asset, which limits their operational value as a complementary technology that can be coordinated with GPS.

This project focused on the development and design of an appropriate reconfigurable patch antenna system that will serve as a waypoint for communication between the wireless Android receivers and the central computational unit utilizing the MUSIC algorithm, which will also operate as an information server for the phone to pull DOA information from. The antenna is designed to 1) be compatible with the Bluetooth and Wi-Fi frequency band, 2) possess the capability to be altered on-the-fly via user control of the app and 3) contain multiple access points and/or be omni-directional in order to derive DOA for multiple targets regardless of positioning. Extensive measures will be required in the design and fabrication due to the significance that the antenna plays in the communication of the system.

As well as the design of the reconfigurable antenna system, the project included the development of an Android application that interfaces and controls an access point capable of implementing the MUSIC algorithm to obtain the high-resolution DOA information. Specifically, the application 1) pairs to a MUSIC-equipped system via Bluetooth and Wi-Fi, 2) configures features of the wireless system and MUSIC algorithm (control the antenna configuration, sampling, frequency, etc.), 3) uses this wireless channel for the calculation of the MUSIC spectrum, 4) receive DOA information for the phone and other users paired to the MUSIC system, 5) graphically display this information to show the location of phone/device and other users relative to the location of the system implementing the MUSIC algorithm, and 6) port this information to a remote server.

CHAPTER II

COMPONENTS AND METHODS

The methods of development in this work were selected to facilitate the rapid development and prototyping of this project. The primary portions of the system can be broken down in three ways: the mobile application, the central basetation and the tracking antenna. The Android operating system was selected as the mobile application's platform of development because of its expansive, open-source programming community and well established developmental language, Java. The central basestation was implemented through the National Instruments LabVIEW environment and operated as the central link for the entire system. LabVIEW was chosen due to its ease of integration across many platforms and standards as well as its intuitive graphical programming interface thus ensuring a quick development of the server. A reconfigurable cross-dipole microstrip patch antenna, as shown in the figure on p.11 was chosen as a good candidate to act as the array elements for the tracking antenna. This antenna would serve well for experimental purposes because it offers relative ease in design, fabrication and reconfiguration.

Mobile application development

The primary goal of this work was to establish a secure communication channel between the user and tracking station in order to provide location information in areas where GPS coverage was incapable or infeasible of being utilized. The Android mobile platform was chosen for several key reasons: it is supported by an extensive open-source community, it is developed through a standard programming language, it is prominent in commercial and industrial markets allowing for straightforward propagation of this work into these sectors, and it is capable of utilizing well-established signal standards (e.g. Bluetooth, Wi-Fi and GPS) which will aid in the function and testing of the system. The desired functions of the application included establishing a dedicated communication link, receiving and storing the DOA information for all the current users, and displaying the information in an easy-to-use manner.

The application was designed to require minimal computational effort by the CPU in order to allow for extended use of the application and can be primarily broken into two phases: the connecting phase and the listening phase. Fig. 1 shows the generic functional flow of the mobile application as it interfaces with the central basestation. Once the application has been started and the Bluetooth interface has been initiated the app waits on the user to prompt for a connection to the central basestation. This ensures that the user intends to communicate with the system and will not be tracked without their knowledge and consent. The mobile device instantiates a connection thread to communicate with the basestation through the multipurpose RFCOMM Bluetooth socket. RFCOMM is a specific Bluetooth protocol that is designed to emulate a serial port, modeled after RS-232. This straightforward implementation of data transfer allows for uncomplicated data transfer between the mobile device and the basestation. A unique connection is established between the mobile device and basestation once the request has been made.

The application then enters "listening mode" and starts a dedicated thread used for listening to the open Bluetooth socket. This thread runs in the background of the phone and must be terminated manually in order to close the connection. The user interface is updated as new DOA information is received and will only halt when the application has received asynchronous input from the user. The application waits for user input in parallel with the Bluetooth thread and does will not change its current state until the signal is lost or it receives input. The user has several options from this state: change the polarization of

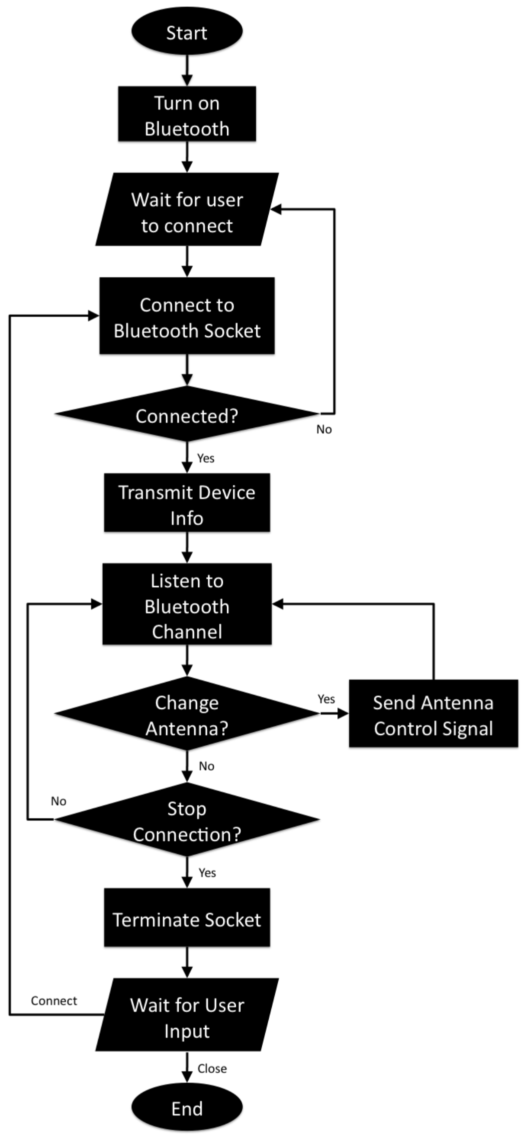

Figure 1. Flow of application state as it establishes and maintains a connection.

the antenna, stop the connection, reconnect to the basestation and close the app. The user changes the polarization of the antenna by selecting the “Antenna Configuration” button as shown in Fig. 2. This will pause the listening thread of the channel and send a command to the central basestation to change the polarization. The user can also terminate the socket connection with the basestation. This is a required function in order to stop communicating with the basestation. This background thread was implemented so the user could access other functions on the phone without losing the input stream of DOA information. The user also has the capability of reconnecting to the server in case the signal was lost or the socket was terminated prematurely. This is operated through the options menu. Finally, the user can close the front screen of the app and not display the information. As mentioned previously, this will not end the data transmission but will still allow the user to access other features of the phone.

Central basestation

The central basestation was designed and established as the principal server for the entire system in order to connect the various peripheral devices and pass data and controls between them. National Instruments’ LabVIEW graphical programming language was selected as the means in which to design and implement the basestation due to its simple programming interface and compatibility with a variety of other software and standards. The functions of the basestation include: managing connections to the mobile devices and relaying data to them, maintaining a DOA and MAC address database and passing control signals and feedback from peripheral devices.

LabVIEW offers a simple and intuitive interface for connecting to Bluetooth devices through the previously mentioned RFCOMM Bluetooth socket as shown in Fig. 3. The

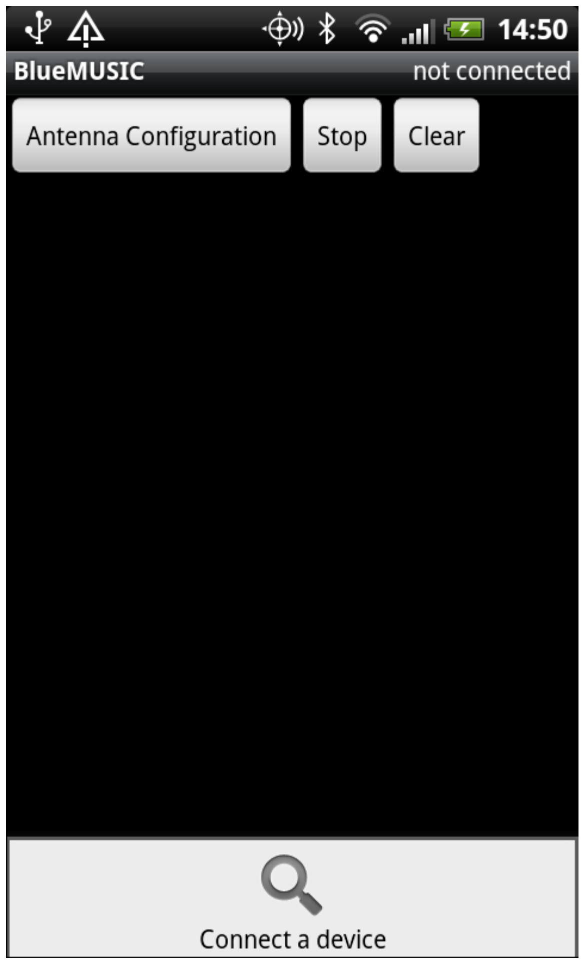

Figure 2. The main screen for the connection app. From here all controls can be implemented.

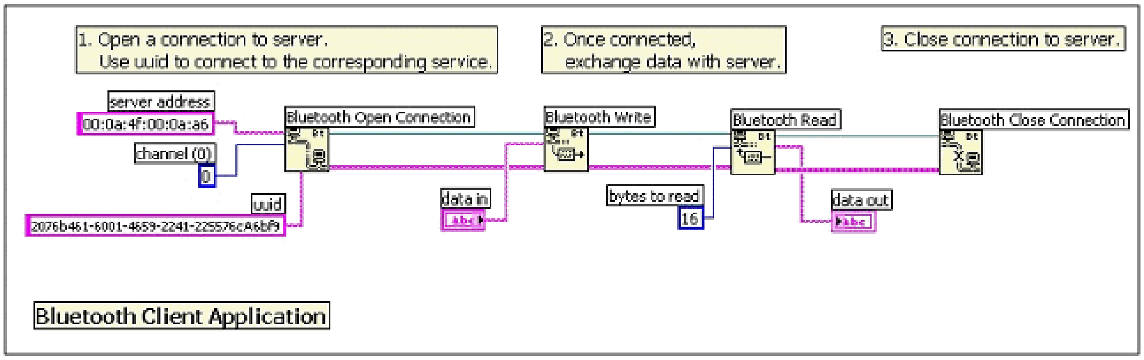

Figure 3. National Instruments example of Bluetooth Client.

ability to establish a Bluetooth socket and ongoing connection with the mobile devices was easily implemented with some modifications to the example. The unique MAC address is added to the MAC-DOA database once a connection is established in order to identify each phone individually and match them to their respective DOA information. Communication between the basestation and mobile device is achieved in LabVIEW through the implementation of a *while-loop* that constantly checks for a message to read and then writes the updated DOA information to the device. In this thread the control signal for the antenna configuration is read and then passed to a subVI for processing. The read VI will timeout if no control signal has been sent and the process will automatically jump to transmitting the MAC-DOA information. The loop will then reiterate infinitely until the signal is lost or the mobile device closes the socket.

Maintenance of the DOA-MAC database was performed by a subVI in the main program for the central basestation. Each MAC address is stored in the database as the phone initiates a connection. This MAC is then passed to a separate VI that is responsible for determining the DOA information. The DOA VI will return the specific DOA information regarding the particular device identified by the MAC address and this information will be stored in the MAC-DOA database. This subVI will continue to operate until the mobile

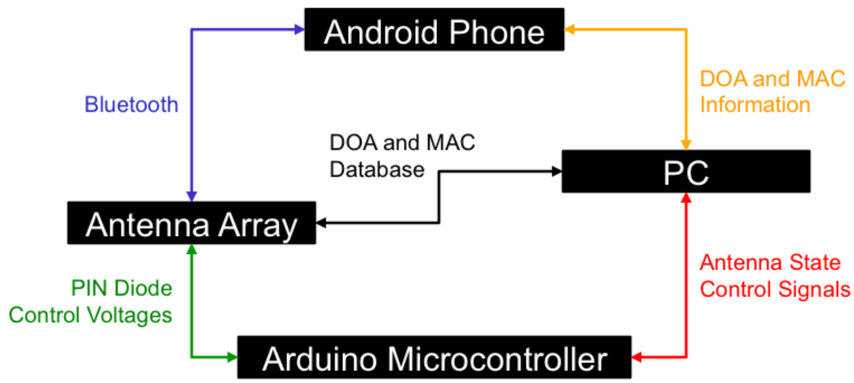

Figure 4. System level data flow diagram.

device is determined to be no longer connected to the central basestation, at which point the MAC-DOA information will be removed from the database.

The central basestation also operates as the path for all communication and control between the peripheral devices of the system as seen in Fig. 4. The only control at the time of this research was the polarization of the DOA antenna array (e.g. vertical or horizontal); however, any number of controls could be added and would only be limited by the desired functions of the system. In order to change the polarization of the antenna arrays the user operated a button on the user interface that sent a control signal through the Bluetooth channel and then interpreted by a subVI in the connection thread of the basestation. This signal would change the output voltage of the Arduino microcontroller that will change the antenna polarization. This function will be discussed later in this paper.

Antenna design and function

A reconfigurable cross-dipole microstrip patch antenna, as shown in Fig. 5, was chosen as a good candidate to act as both a transmitter and receiver for such a system. This antenna would serve well for experimental purposes because it offers relative ease in design, fabrication, and reconfiguration.

Fig. 5 shows the reconfigurable cross-dipole antenna from a 2-D top view. It consists of two microstrip dipoles oriented perpendicular to each other over a ground plane. The dipoles are nominally half-wavelength and produce two orthogonal polarizations. The dipoles are segmented into five sections separated by small gaps. The five sections are connected using RF diodes to provide continuity. The diodes are used to turn the dipoles on and off such that polarization of the antenna is controlled actively by the system. Both dipoles are fed at the center section using a coaxial probe. By utilizing orthogonal polarizations a kind of "spatial filter" is created. Assuming that some future devices will have either vertically or horizontally polarized antennas the construction of the receiving antenna allows the system to pre-select from either the vertically or horizontally polarized devices.

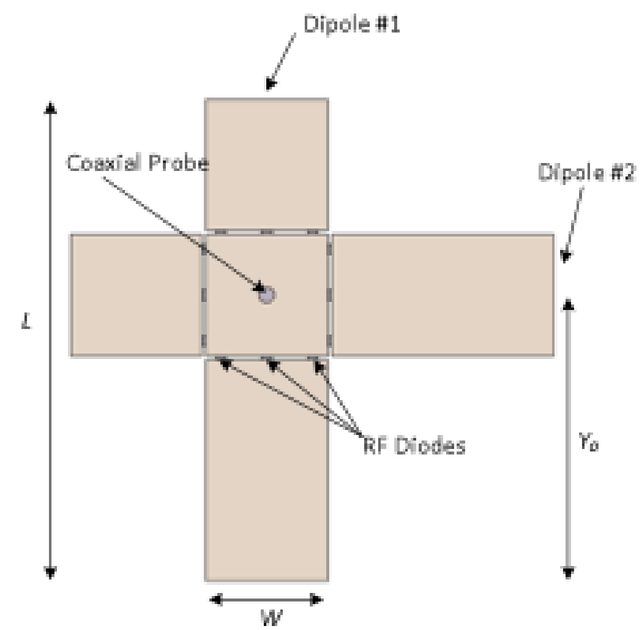

Figure 5. Reconfigurable cross-dipole antenna shown from a 2-D top view.

CHAPTER III

COMPLETED DESIGN, RESULTS AND FUTURE WORK

Completed design and results

This section presents the completed physical models and interactions the client and server sub-components as well as the simulation data acquired for the cross-dipole antenna. Results are presented which correspond to design parameters in Chapter II.

Cross-dipole simulation

The antenna design is based on balancing the width W, length L, and inset Y_0 to match its impedance to the coaxial feed and tune its fundamental resonance to desired frequency as performed in the work done in [5]. The length L of each arm is roughly a half-wavelength. This dimension is used to change the resonant frequency. Changing the inset Y_0 moves the feed point along the length of each arm to tune the input impedance while adjusting the width W also affects the impedance.

The use of diodes to turn the dipoles on and off requires a DC biasing scheme. Fig. 6 provides a passive conceptual-level schematic of the antenna in which the patch is divided into five different sections. The middle section is biased with a voltage V_{DC} using the inner conductor of the coaxial line (i.e. the probe). The outer four sections are connected to DC ground. With the diodes oriented as shown in the figure, the operation of the antenna is as follows. When V_{DC} is sufficiently positive, the vertical dipole is turned on, and the horizontal dipole is turned off. This is referred to as “state 1”. When V_{DC} is sufficiently negative, the horizontal dipole is turned on, and the vertical dipole is turned off. This is referred to as “state 2”.

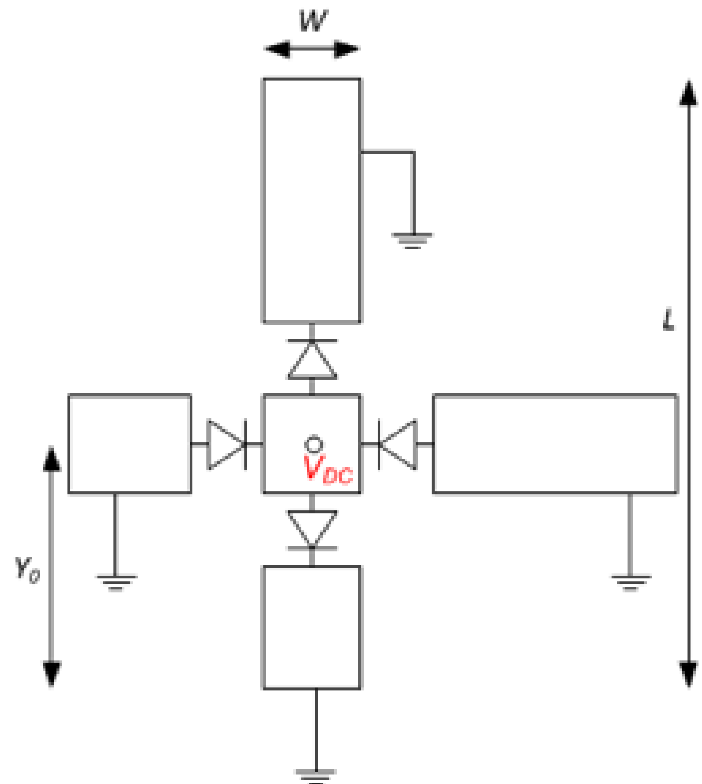

Figure 6. Conceptual-level schematic of reconfigurable cross-dipole antenna depicting the passive DC biasing scheme.

An active DC biasing scheme has been developed in order to allow individual control of arms independently. This will allows the system to use 3 states instead of the previously listed 2. “State 1” and “state 2” will remain as the vertical and horizontal polarization states, respectively; however, an active scheme will allow a third state in which both polarizations are turned on. This scheme allows for a kind of “spatial filtering,” allowing only those of the similar polarization to be “seen” by the antenna. The conceptual level schematic is shown in Fig. 7.

In the practical implementation of the passive DC biasing scheme, the ground plane is connected to DC ground. Additionally, the outer conductor of the coaxial feed is DC-grounded. In this way, the bias voltages can simply be taken from an external circuit

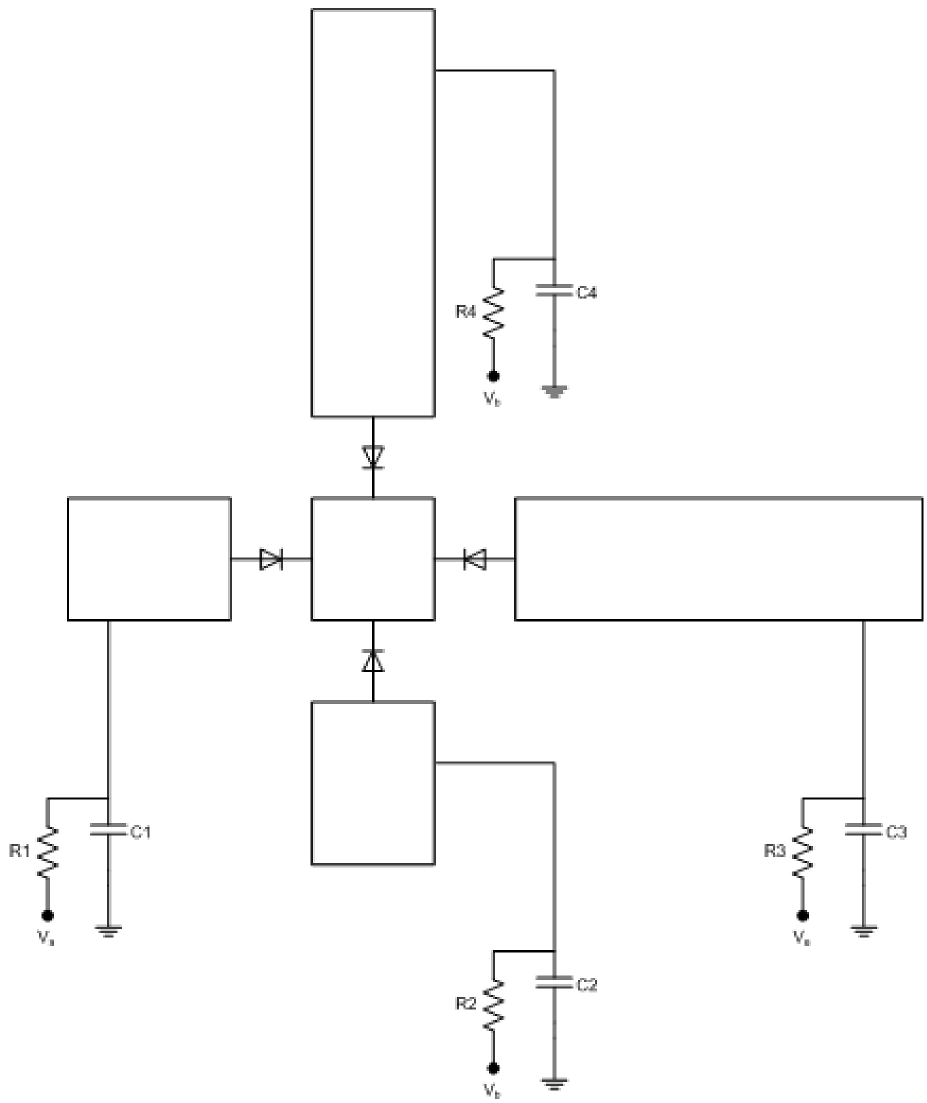

Figure 7. Conceptual-level schematic of reconfigurable cross-dipole antenna depicting the active DC biasing scheme.

and superimposed onto the coaxial input. This configuration requires that the four DC-grounded microstrip sections of the dipoles be DC-connected to the ground plane. To do this, four microstrip bias arms one extending from each portion of the dipole are included in the design with vias to ground at the end of the arms. These are thin, high impedance lines that are designed at roughly a quarter-wavelength to imitate an RF open circuit but simultaneously provided DC connectivity to ground. Fig. 8 shows a top view of the antenna with these bias arms.

Similarly, for the practical implementation of the active DC biasing scheme, the ground plane is connected to DC ground as well as the outer conductor of the coaxial feed. For the active control case instead of the bias arm connected directly to ground by via there is a resistor and capacitor placed on the substrate in parallel. The high resistance resistor will be connected to an analog output of the Arduino microcontroller to shift the voltage for the arm itself to engage or disengage the PIN diodes. The parallel capacitor will halt any leaking current from running straight to the ground from the DC source. As mentioned above, this will allow active control of the arms independent of the center patch voltage. For this case, the voltage of the center patch does not need to be DC biased.

Several iterations of antenna design were simulated in Ansoft HFSS to determine RF performance. Fig. 9 shows the CAD model of the best simulated designs without biasing arms. The displayed parameters are W=5mm, length L=40.7mm, and Y_0=17.375mm. This antenna will be considered to be the desired result for the system for the purposes of this paper. Biasing arms will not be taken into account at this time because they will be nominally a quarter wavelength and thus an open circuit electromagnetically, contributing negligibly to the RF characteristics of the antenna. The substrate has a dielectric constant $\varepsilon_r = 2.2$ and a thickness $h = 0.062 mils$ (corresponding to a Rogers duroid 5880 substrate).

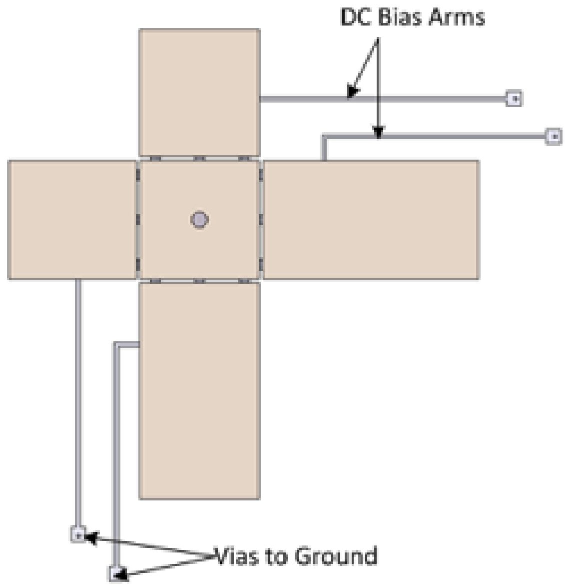

Figure 8. Simulated layout (top view) with DC bias arms. Each bias arm has a via to ground.

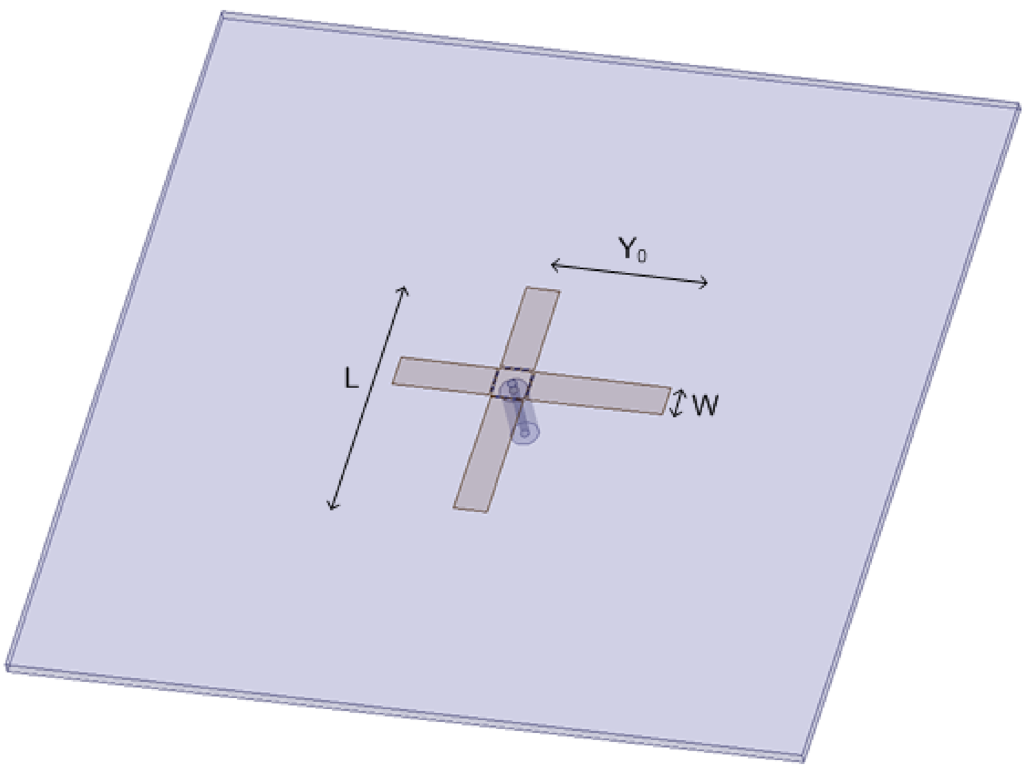

Figure 9. HFSS simulated antenna design.

No DC voltages and no diodes were used in the simulation. Instead, the diodes were approximated with lumped elements corresponding to the forward and reverse bias conditions of the diodes. Specifically, in the forward bias case, a resistance of 2Ω and an inductance of 0.7 nH were used. In the reverse bias case, a resistance of 10kΩ and an inductance of 0.7 nH were used. These values were taken from the data sheet of the RF diodes used for fabrication, SMP1345-079F PIN diodes.

Fig. 10 shows the simulated VSWR vs. frequency for the different configurations of the antenna after optimization. Because of the symmetry of the single polarizations (i.e. either vertical or horizontal) only one is shown with the other being dual polarization. It is clear from the figures that the dual polarization does not match perfectly at the frequency band that the single polarizations do; however, all configurations have good VSWR responses in the 2.4-2.5 GHz ISM band. This allows the antenna to be feasible for this iteration of the experimental proof-of-concept process. Further modifications will be performed to center the responses around 2.45 GHz.

It can be seen from Fig. 11 that neither states are an ideal match for the input impedance. It was determined for the purposes of this research that an even distribution of both configurations (single and dual) on either side of a perfect match on the Smith chart was acceptable. The small frequency shift in the two VSWR plots, as shown in Fig. 10 between the different configurations was determined to be a minor issue due to the fact that the VSWR is in an acceptable range across the desired transmission band of 2.4-2.5 GHz.

Fig. 12 shows the simulated radiation patterns of the antenna. It is evident that the patterns for the dual polarization state show a strong similarity to the single polarization states in both the phi sweep. In the theta sweep it is seen that the dual polarization field is still propagating while the single polarization is turned off. Measured radiation patterns were

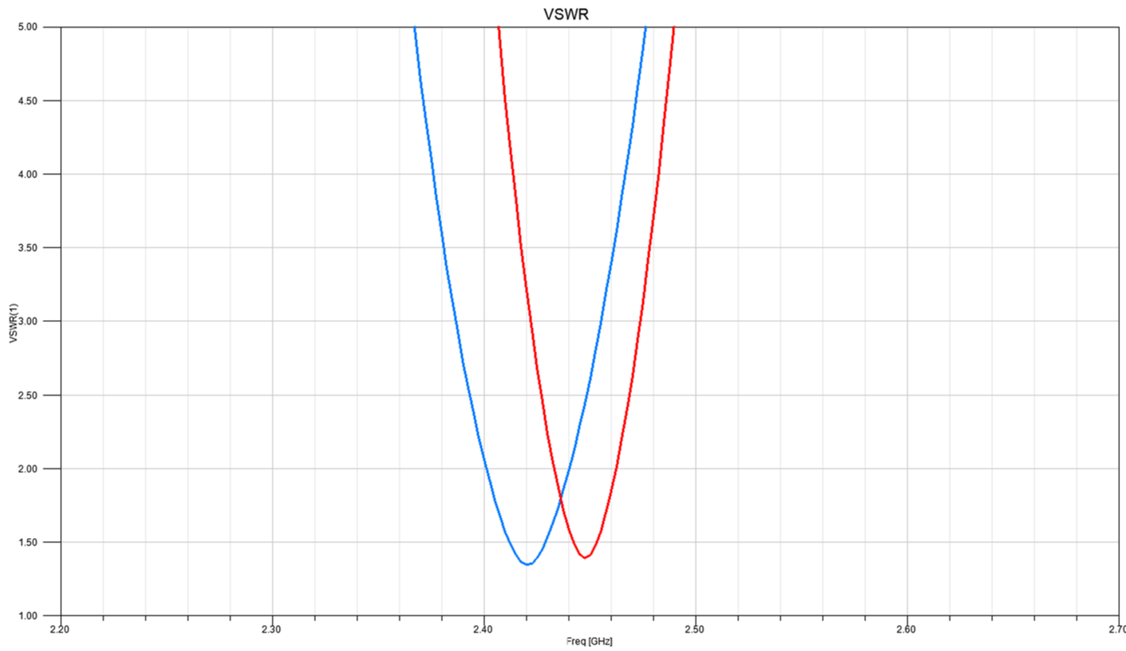

Figure 10. Simulated VSWR measurements for state 1 & state 3 antenna configurations with state 2 not shown.

not obtained due to the lack of an anechoic chamber suited for the operating frequency. However, such measured data was not deemed necessary at this time.

Mobile phone interface

The phone interface is designed to allow the user to receive the DOA information in a useful way and to have control over the current polarization scheme of the cross-dipole array. The interface for the primary screen is shown in Fig. 13. The primary screen simply shows a list of users currently connected to the central basestation and their respective DOA information. This list view then can be used to select a specific user to bring up a screen where more information is given about the respective user in the future. More detailed information can be shown here such as total connection time, phone diagnostics and personal user information.

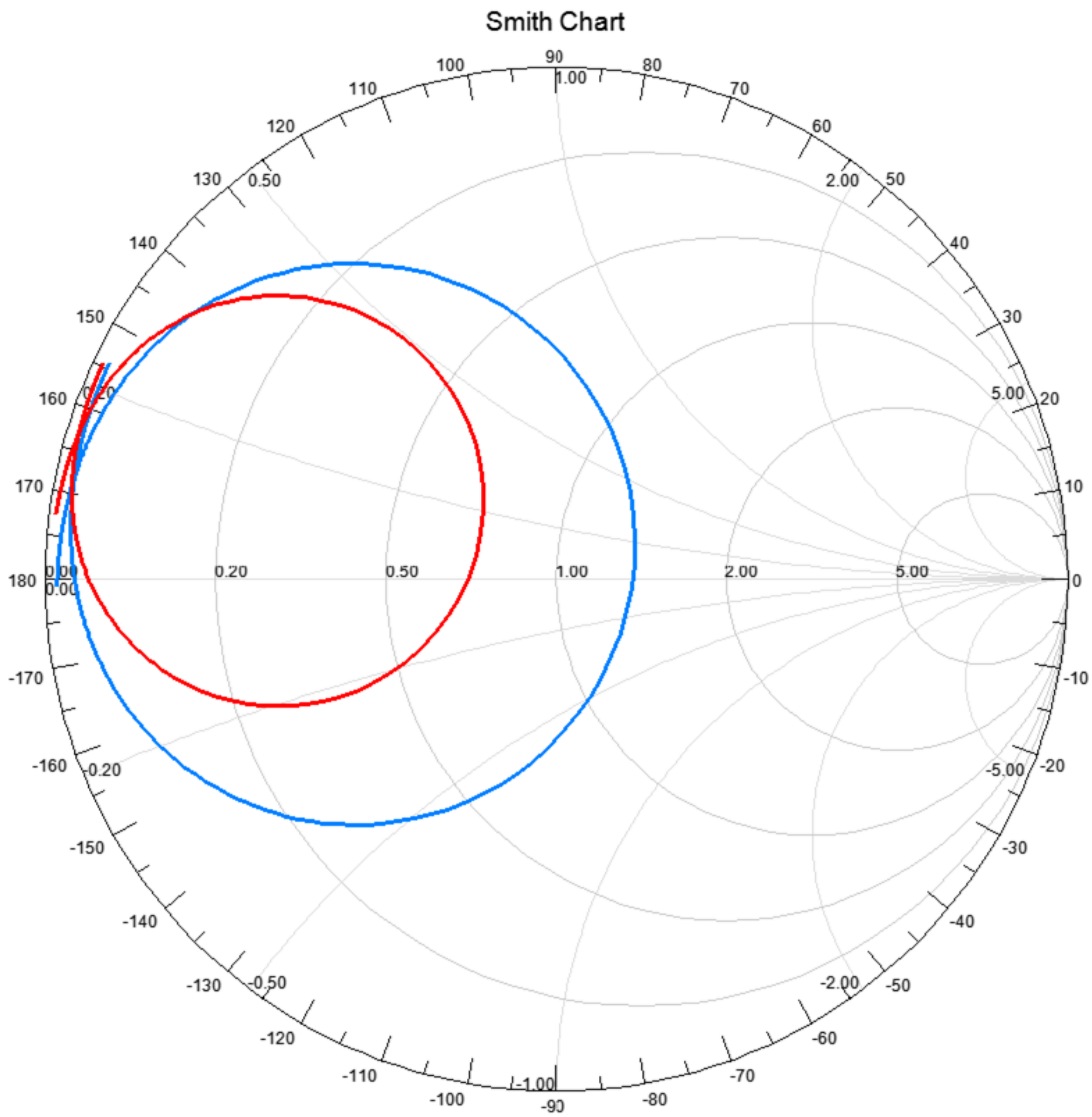

Figure 11. Smith chart for single polarization impedance (blue) and dual polarization impedance (red).

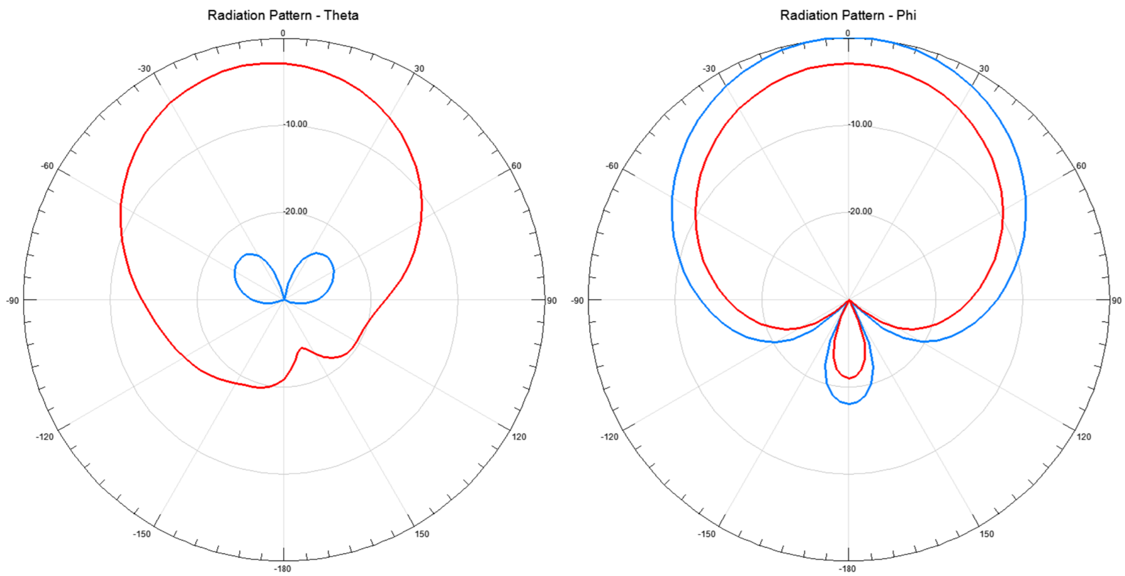

Figure 12. Radiation patterns for single and dual polarizations. Note there is a small difference in the dual-polarization field as compared to the single-polarization field.

The controls of the application will be managed through the menu options button. The user is able to select a variety of controls from the menu: connect to the basestation, change the current polarization or the cross-dipole array, enter a "compass" directional view or stop connection with the basestation. The user must connect to the basestation to start the tracking process, which is implemented by selecting the "Connect" button from the menu options shown in Fig. 2. The user will be able to cycle through the different polarizations of the cross-dipole array by selecting the "Change Polarization" button. This was implemented for all users in this research; however, in a more developed setting this would most likely be a control for a moderator or superuser to spatially filter and targets that were being tracked. The "Compass" button initiates a compass view that the user is able to use to navigate around relative to the central basetation. The compass, which is based on the phones internal compass utility, is recalibrated for the top of the phone to be oriented relative to the antenna array as shown in Fig. 14. The phone then is able to add or subtract 180° from the tracked angle to always point in the direction of the basestation.

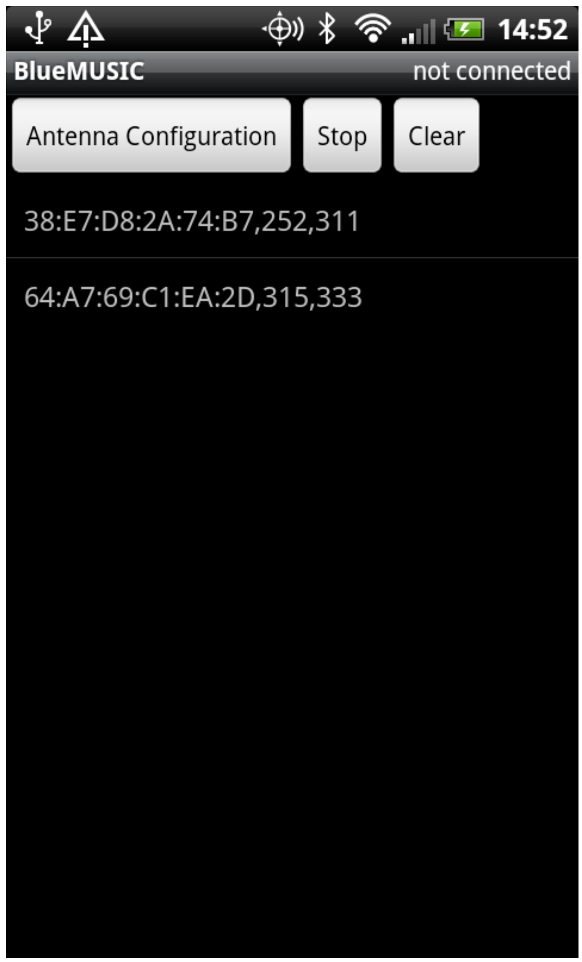

Figure 13. Main screen displaying the current list of connected users to the system.

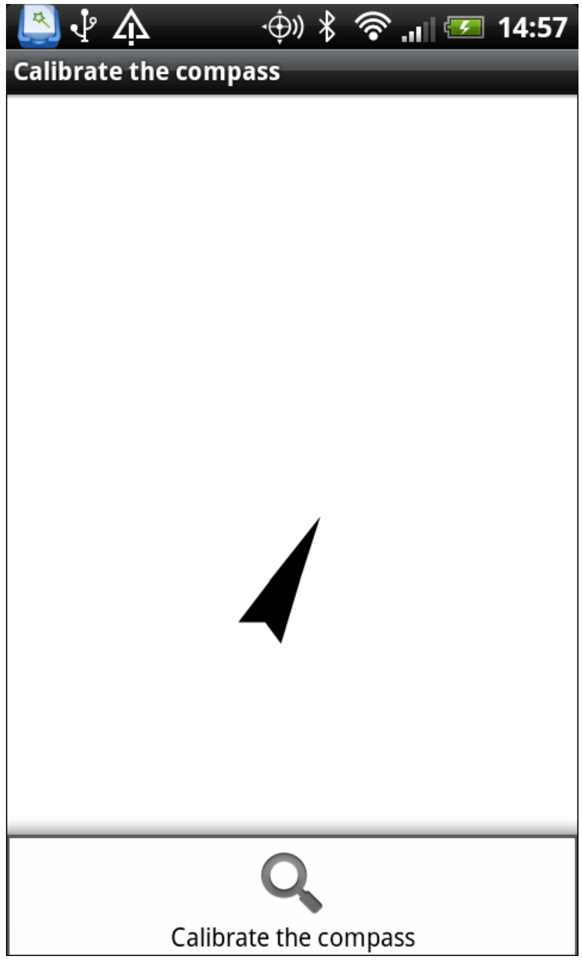

Figure 14. The compass screen used for simple angular navigation as a proof-of-concept.

Finally, the “Stop Connection” button will terminate the connection the phone has with the central basestation and remove the respective device from the list of currently tracked targets in the MAC-DOA database.

Central basestation interface

The central basestation interface as shown in Fig. 15 is provided primarily as a means to allow a user to troubleshoot and debug the system. Displayed would be the current phones connected to the system as well as the incoming and outgoing information on the Bluetooth channels. Any error information that might occur on the server side is also displayed at the bottom of the screen.

Whenever a new phone connects to the system it is tallied and placed into the phone array shown on the left of Fig. 15 and detailed in Fig. 16. The device list array immediately to the right displays the MAC address of the devices connected as well as the DOA information taken from the DOA algorithm. This array is the actual information transmitted to the phones and then is displayed on the phone as Fig. 13. Currently, as each phone connects it begins a parallel thread that runs on the server, detailed in Fig. 17. This method ensures complete and uninterrupted communication between the client and server without having to worry about multiple phones "talking" over each other.

Future work

The future work of this project will be primarily concerned with implementing a full-scale system, which will be performed in several key ways. These include continued design of a better performing antenna, a more robust sever side system, and full-scale testing.

First, the continued design of an appropriate antenna will need to be considered. Several implementations have already been considered and are ready to be tested. These primarily include methods to reduce capacitance across the gaps of the different parts of the antenna. The end goal of continuing to design a better performing antenna will be to center the VSWR spikes more closely. This will allow the antenna to be much easier to operate and account for as the full system is pieced together.

Currently, phones must be paired with the Windows 7 operating system on the server prior to connecting to it through the LabVIEW program. This is not an issue currently; however,

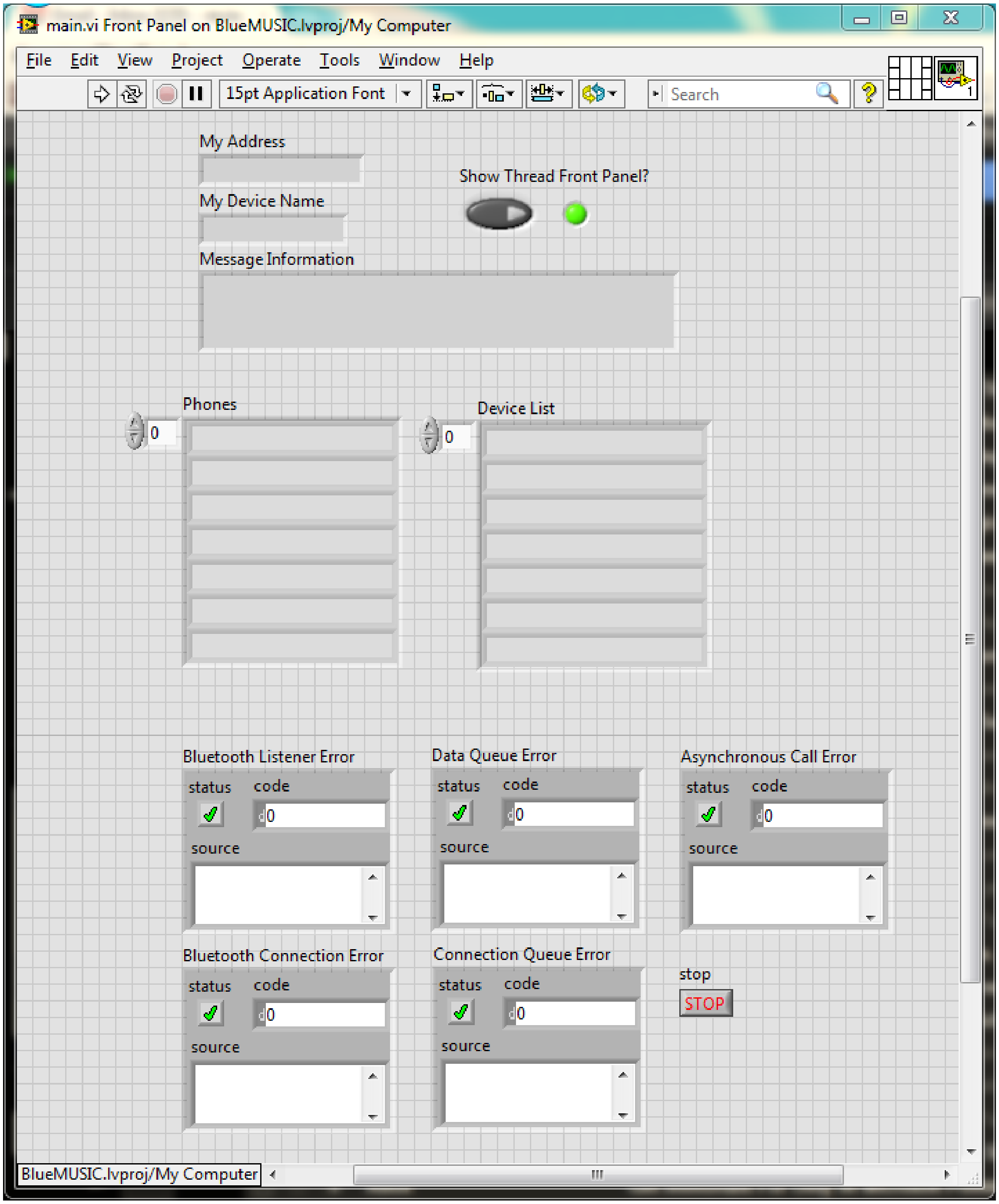

Figure 15. Server side interface. There is no direction control for the server; however, current phones being tracked and other system information can be seen.

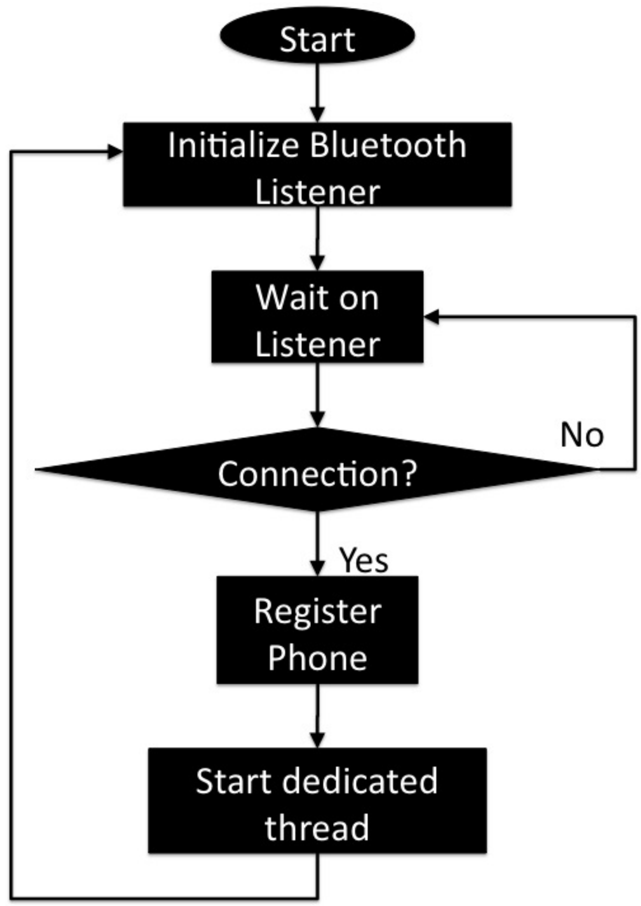

Figure 16. Process executed by the server to allow a phone to connect. This method runs indefinitely until the server is stopped.

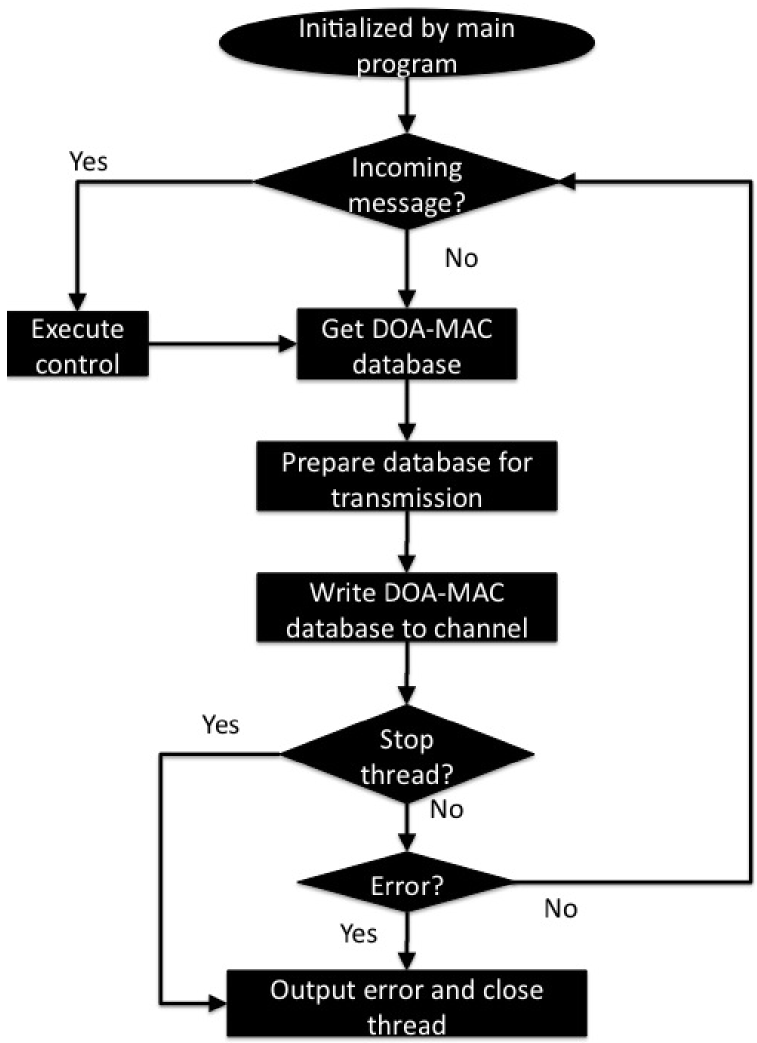

Figure 17. Process executed to communicate with the phone. This method runs indefinitely until the channel is closed by either the server or the phone.

to produce a more commercially viable system a dynamic pairing process will need to be implemented. This could be possible through LabVIEW's .NET compatibility modules as performed in [6] with Nintendo Wii remote controllers.

Full-scale testing of the system will need to be performed. This includes the integration of both the DOA algorithm and the communication system described above. At the time of this paper, client to server communication had been established; however, with the antenna still in the simulation phase it was impossible to test the antenna configuration switching mechanism. Nonetheless, work was performed to make this step relatively simple.

CHAPTER IV

CONCLUSION

Ease of integration was necessary to realize a working prototype as soon as possible. With this philosophy in mind, the project set out to use standard technology including Bluetooth, Android OS, LabVIEW and Arduino in innovative ways. With steady determination throughout the course of this project a successful communication link has been established to transmit DOA information efficiently between a tracked phone and the tracking station. Each piece of the system came together to produce this final result, each requiring a diversified method of tackling individual module issues as they arose.

The phone interface required an understanding of basic object oriented programming. With much help from the wide open-source Android community each step of the system was tackled successfully from opening Bluetooth protocol sockets for data transmission to basic phone functionality (e.g. keyboard interface, etc.). The end result was an application infrastructure that can be easily expanded upon in the future for commercial, industrial or military uses.

The basestation/server programming again required a new kind of understanding of programming. LabVIEW lends itself naturally to sequential thinking and programming and in order to develop parallel threads required learning newly designed parallel processing features of LabVIEW 2011 that had previously been unavailable. Communication with the LabVIEW developers, National Instruments, was key in ensuring the server side operated as desired. As the center of all data and control flow designing an efficient server system was key and was realized in this project.

Because Bluetooth and Arduino are readily available standards well used throughout the rest of the world they were natural to integrate into the project. National Instruments had previously released an Arduino driver kit allowing a very simple interface to be developed in order to communicate with the microcontroller. Similarly, Bluetooth had well established application programming interfaces for both LabVIEW and Android. Having both platforms share a similar method of communication facilitated a quick turn around for the production a communication link between server and client.

Due to the ease of integration off all the software a significant portion of the research was dedicated to the design of a functioning cross-dipole antenna. Utilizing the parametric sweeping function of Ansoft HFSS was critical to the design of the antenna because the antenna can be realized as two separate antennas packaged in a stacked fashion. With minor exceptions, this rendered standard equations for a patch antenna nearly unusable. Thus an experimental research design was used in order to determine the affect of each parameter W, L and Y_0 on the antenna. Through an intuitive understanding of electromagnetics, data interpolation and parametric experimentation an antenna was successfully designed to meet the criteria of being matched well and operating in the desired bandwidth.

As an end result, the successful implementation of the communication side of the system gives promise to a series of mobile applications that will be able to help navigate people around large indoor areas. These may include multistory buildings, sports stadiums, airports and in reality any building. Wherever people may be lost there is a functionality for a system like this.

REFERENCES

[1] A. Broumandan, J. Nielsen, and G. Lachapelle, "Enhanced detection performance of indoor gnss signals based on synthetic aperture," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 2711 –2724, July 2010.

[2] Y. Chen, R. Chen, X. Chen, W. Chen, and Q. Wang, "Wearable electromyography sensor based outdoor-indoor seamless pedestrian navigation using motion recognition method," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference*, (Guimaraes, Portugal), pp. 1 –9, Sept. 2011.

[3] J. Link, P. Smith, N. Viol, and K. Wehrle, "Footpath: Accurate map-based indoor navigation using smartphones," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference*, (Guimaraes, Portugal), pp. 1 –8, Sept. 2011.

[4] V. Moghtadaiee, A. Dempster, and S. Lim, "Indoor localization using fm radio signals: A fingerprinting approach," in *Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference*, (Guimaraes, Portugal), pp. 1 –7, Sept. 2011.

[5] D. L. Rolando, Hung D. Ly, Tie Liu, G. H. Huff, J. P. Erskine, J. A. Joachim, and J. B. Yang, "Channel-based wireless encryption using reconfigurable antennas." presented at the International Union of Radio Science/Union Radio Scientifique Internationale Conference, Boulder, CO, 2012.

[6] "Community: LabVIEW interface to a Wii Remote ("Wiimote")." Internet: https://decibel.ni.com/content/docs/DOC-1353, Aug. 6, 2011 [Feb. 12, 2012].

CONTACT INFORMATION

Name:	Daniel Lee Revier
	c/o Dr. Gregory Huff
Address:	Department of Electrical and Computer Engineering, Texas A&M University, 214 Zachry Engineering Center, TAMU 3128 College Station, TX 77843
Email Address:	revier.daniel@gmail.com
Education:	B.S., Electrical Engineering, Texas A&M University, 2012