RADIUS

Richard W. Brown, EE, Titus Rwantare, CSE, Cormac Kennedy, EE, and Ibrahim Tahoun, CSE

***Abstract*—At one point, we have all felt the pain and frustration of losing a valuable personal item. The Radius Bluetooth tracking system is a solution that not only enables you to find your possessions at the push of a button, but also prevents losing them in the first place. Consisting of a Bluetooth tracking tag and an Android application, Radius establishes a perimeter around you and your devices and alerts you when paired devices exceed an allowable range, drastically reducing the amount of time and money it takes to find or replace lost items. Radius does not just solve the lost item problem; it removes the problem entirely.**

I. Introduction

Recovering a misplaced personal item can be extremely resource intensive. Whether you are wasting time looking for a lost item or spending hard-earned money to replace it, we can agree that the situation could be improved if the item had not been misplaced in the first place. The Radius Bluetooth tracking system prevents lost items by providing real time alerts via an Android application and Bluetooth tracking tag when an item is at risk of being left behind.

A. Significance

People all over the world lose things every single day, and as the pace of our lives continues to increase, so does the number of lost items we experience. James Gleick writes in his book *Faster*, "We are spending about 16 minutes of each day looking for misplaced items which is almost a year of our entire life searching for lost possessions," [1]. According to the United States census bureau, the median annual income of a United States citizen is $31,177 [2]. This means that an average American has the potential to lose over $30,000 worth of time looking for lost items in their lifetime, and this figure is not even factoring in the cost of replacing lost items.

B. Context and Existing Products

There are many Bluetooth tracking technologies available on the market, the most successful being the Tile Bluetooth tracker. The Tile tracking system consists of a Bluetooth tracking tag, and mobile phone application. By attaching a Tile tracking tag to an item, you can find it by sound, or by seeing its last known location on a map [9]. Another prominent Bluetooth tracking company is Chipolo, which offers the exact same exact functionality as Tile in a different aesthetic [10]. Both companies allow you to find your lost items, but neither of them prevents losing an item in the first place. Our Bluetooth tracking system not only provides the same functionalities as Chipolo and Tile, but goes one step further to prevent losing an item before it can be left behind. By constantly checking to make sure you have not been separated from your items, our tracking system is able to inform the user in real-time that they are about to lose an item in order to avoid the need to look for it at all.

C. Societal Impacts

Everyone has experienced losing an item, whether it is a mother who has lost her child in a grocery store, or a college student who has lost their cellphone in their dorm room; it is rare to find a person who would not admit to losing a prized possession. Since the problem is so widespread, the solution should be accessible to everyone. Accessibility plays a huge role in how our product is designed and implemented. A convenient Android application allows anyone with a smartphone to download the application, and anyone with more than twenty dollars the opportunity to never lose a personal item (final product = ~$25). The goal of Radius is just that, to give everyone the opportunity to keep what they value, in a way that is economic, convenient, and accessible.

D. Requirements Analysis and Specifications

Specifications	Tracking Tag
1. I/O	Will make noise when button on phone is pushed
3. I/O	Will make noise when paired devices exceed allowable range
4. Power	Battery life of 1-3 months utilizing 3V Coin Cell battery
5. Size	Area of 25 cm^2
6. Size	Thickness of 5 mm

Specifications	Phone Application / Android Device
7. I/O	Will ring when button on tag is pushed
8. I/O	Will ring when paired devices exceed allowable range
9. Performance	Capable of running high and low priority modes

Figure 1: Requirements and Specifications

Requirements and specifications for our final product were based off our current development board and other competitor products. The size of the tag being set to 50x50mm is approximately half the size of a credit card, allowing for seamless placement inside a wallet, on a keychain, or attached to an electronic device (phone, tablet, laptop). This size makes our design comparable in size to existing products. The thickness of the board is governed by our thickest component, the battery. The 5mm height requirement will allow us to place a CR2032 3V coin cell on our board.

Our power specification was calculated by analyzing the CR2032 coin cell battery life along with the DWM1001 development board's power consumption while active and in sleep mode. We found the weighted-average current by using the equation below:

$$I_{AVG} = \frac{I_{ACTIVE}t_{ACTIVE} + I_{SLEEP}t_{SLEEP}}{t_{ACTIVE} + t_{SLEEP}} \times \frac{16\ hrs}{24\ hrs}$$

The 16hrs/24hrs adjusted the result for the number of hours in a day we expect the system to be in use. With a 0.2 Hz (5 seconds) duty cycle and a fixed amount of time for the active ranging process of $t_{ACTIVE} = 32.6ms$, $t_{SLEEP} = 5000 - 32.6 = 4967.4ms$. With an active current of $I_{ACTIVE} = 14mA$ and a sleep current $I_{SLEEP} = 0.013mA$, we get $I_{AVG} = 0.07mA$.

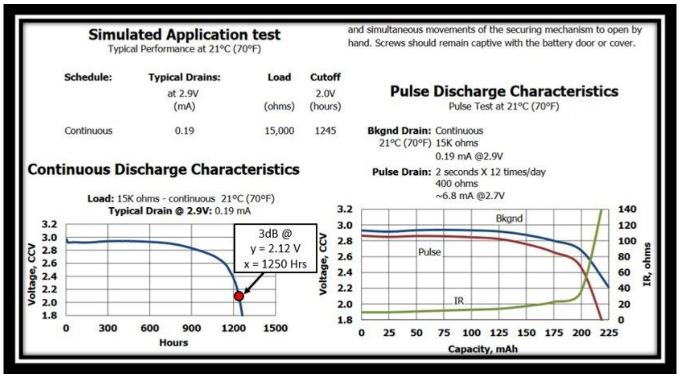


Figure 2: Power Consumption calculations

Using the Energizer CR2025 datasheet above, we multiplied the battery lifetime of 1250hrs by the fraction of the constant current used in Energizer's test case. The equation is as follows:

$$t_{EXPECTED} = t_{TEST\ CASE} \times \frac{I_{TEST\ CASE}}{I_{AVG}}$$

where

$$t_{TEST\ CASE} = 1250\ hrs, \qquad I_{TEST\ CASE} = 0.19\ mA$$

$$t_{EXPECTED} = 1250 \times \frac{0.19}{0.07} = 3419\ hrs = 142\ days$$

We chose 3 months as our maximum lifetime in a low priority mode to give ourselves a small amount of leeway.

Sound produced by the tag will be produced using a piezo buzzer attached to its I/O when triggered by the application on the user's phone. Likewise, a button mounted on the board will cause the phone to ring if the user would like to locate it, or when a tag is out of range.

II. Design

A. Overview

To create a tag that helps people keep their items, we felt that it needed to be small and require minimal maintenance. To achieve this goal, we selected the Bluetooth Low Energy (BLE) protocol. BLE allows us to build small and power-efficient tags. The presence of Bluetooth antennas in modern smartphones decreases the need for one piece of hardware (anchor node). We only intended to make one tracking tag (tag node) and have it interface with the Bluetooth system already in smartphones.We made the switch to UWB for ranging in the second semester for a more accurate ranging method. The benefits and shortcomings of these technologies are further discussed in Appendix Section A. The balance between size, cost, power-draw, availability, and performance was the basis of our decision.

The components represented in our block diagram in Figure 3, are what will be sufficient to create our basic loss prevention system capable of warning users when they are move 5 meters away from the tracking tag (or from the phone with the tracking tag on their person). There may be more components added to the application and board design if our testing deems it to be necessary.

After completing the project to CDR requirements, we were able to satisfy most specifications in our table. The main specification that we were unable to satisfy prior to CDR but continued to improve throughout the end of the semester is the battery life performance of the Radius tag. After feedback from evaluators that incorporating a higher resolution tracking method like UWB would improve the overall performance of

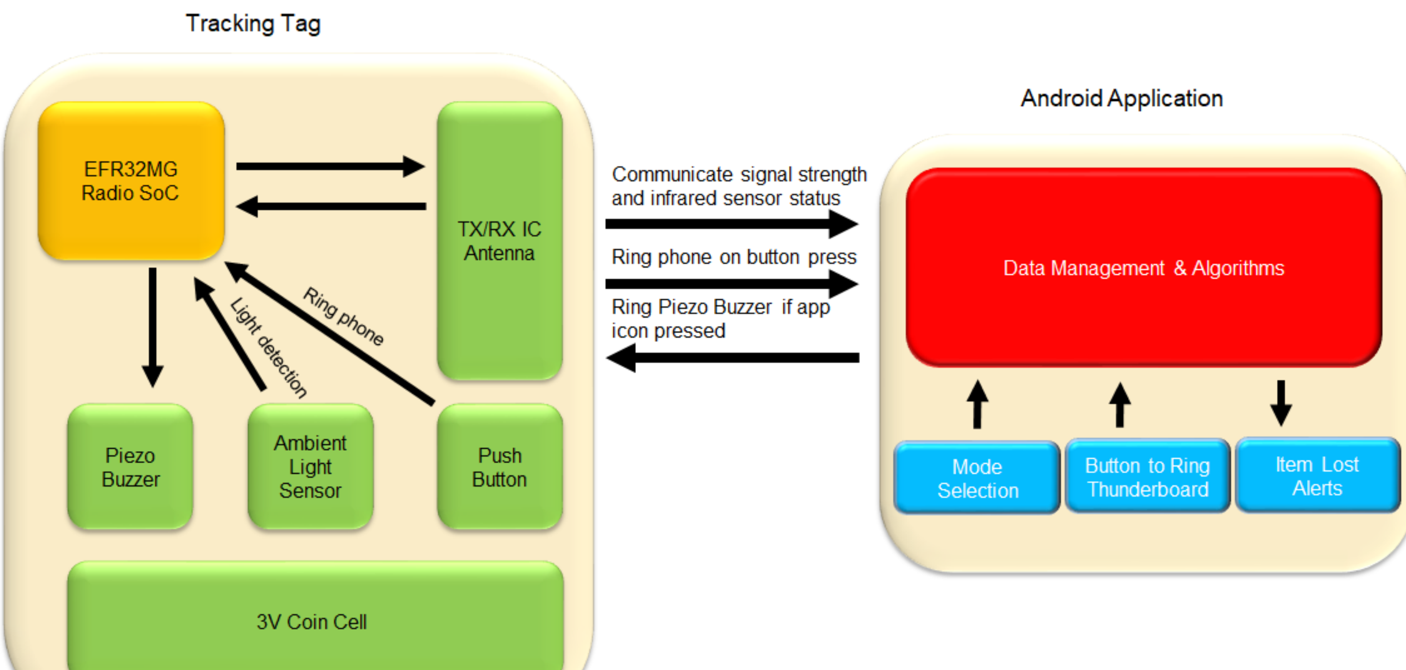


Figure 3: System Block Diagram

tracking, we were faced with a power consumption obstacle. We did not know exactly how to control the low power mode of the module at first which made low power consumption a bit harder to achieve, but ranging accuracy greatly improved. Although we were not able to achieve even a 1 month lifetime by CDR, one team member continued to improve the tracking functionality of the tag to bring overall battery consumption down closer to specification. Satisfying other specifications were all successful regarding tag dimensions and ringing functionality

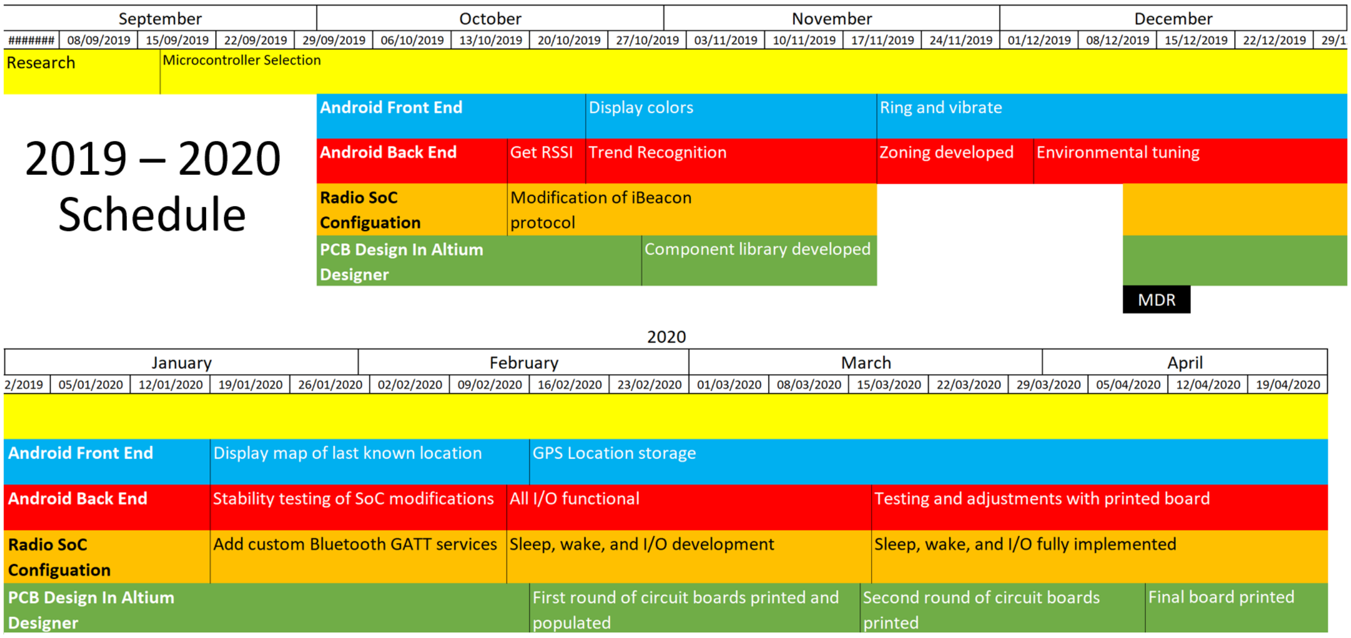


Figure 4: Gantt chart - BLUE: Titus Rwantare RED: Cormac Kennedy ORANGE: Ibrahim Tahoun GREEN: Richard Brown

B. Block 1: Tracking Tag

The tracking tag will contain the components needed to power itself and communicate with a user's devices. Most of the complexity of the tag revolves around operating the EFR32 SoC. This SoC contains the Bluetooth radio whose transmissions' RSSI (Received Signal Strength Indicator) is measured by the phone. This measurement is then input into the logic of the Android application. The Android application uses the RSSI to estimate the proximity of the tracking tag to the Android device.

The RSSI is a measure of the power level of the signal transmitted by the tracking tag as detected at the Android device. A strong Received Signal Strength Indicator (RSSI) means the Android device and tracking tag are in close proximity, while a weak RSSI is indicative of a greater distance between phone and tracking tag. Additionally, the SoC will be programmed to emit an alarm when the signal to the phone is lost, and when the user is manually trying to locate the tracking tag. Messages transmitted from the phone to the tracking tag will enable the CPE-120 piezo-electric buzzer [7] to create an audible noise that will help to inform the user of the tracking tags location. The PCB will also have a ceramic antenna to transmit the Bluetooth signal to usable ranges.

The tracking tag will be powered by a CR2032 3V coin cell. This battery was selected in order to produce a sleek PCB design whilst also providing enough capacity to be able to satisfy the power specification described in Figure 1 (Specification #4). The CR2032, along with the coin cell mount will be the thickest component on our PCB and has been measured to be under 5mm in thickness, which satisfies the thickness specification in section 1, part D of this report (Specification #6).

We opted to alert the user with a CPE-120 piezoelectric buzzer due to the compact size and adequate volume [7]. The buzzer will produce noise if a user explicitly triggers an alert in order to find the item the tag is attached to. Additionally, the tag will emit noise when the connected smartphone goes out of range. The piezo-electric buzzer will be powered by a 3 Volt, 5 kHz square wave that will be produced by the EFR32 [6].

Our PCB will utilize an APDS-9007-020 Ambient Light Sensor which will allow the tracking tag to be able to detect if the tagged object is enclosed in a bag or pocket [8]. The state of being enclosed will result in more frequent signal transmissions, allowing for better system performance with an attenuated signal. For example, if the amount of ambient light sensed by the tag is below a certain threshold, we assume that the tag is being enclosed in a pocket or bag, and that the sensitivity of the application needs to be adjusted to account for the signal attenuation caused by this situation.

A push button on the tracking tag will provide a manual input source on the tag. This input will be used to ring the smartphone if a user wants to locate their phone using the tag. To create a robust tracking tag, the testcases we have considered so far are described in Appendix Section B. The testcases will be key in tuning the behavior of our software in situations where there are obstacles impeding the signal, even though the tagged item is not lost.

C. Block 2: Android Application

The Android application is a crucial component of our setup because it allows the user to interface with the tags. The application will enable the user to ring the tag and allow the phone to be rung by the tag manually. The application hosts the algorithm that also enables the phone and tag to ring automatically when they are far apart, the threshold for this automatic warning will be configurable by the user, they will have the choice between high and low priority items, with higher priority items resulting in an alert at a smaller radius of separation.

To reduce the probability of the phone ringing when the tag is close proximity but with an attenuated signal, the algorithm we have written maintains a short-term trend monitor. Monitoring the trend allows us to determine if the tag gradually got out of range - alert for this, or the tag suddenly went out of range – do not alert for this. In the latter case we assume that a signal blocking obstacle is suddenly introduced, and the tag is still in proximity.

The Android application is built with the Java programming language and utilizes the Android Bluetooth library [11] to interface with the phone’s built-in Bluetooth radio and communicate with the external tag. We utilize this connection to transmit the alert state determined by the software logic to the external tag.

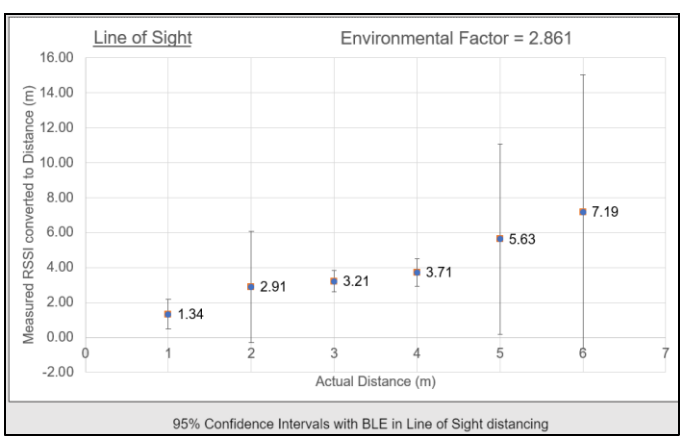
Finally, the app uses the Android location library to log the GPS location at which a tag’s signal was lost in case the user had disabled or ignored the automatic alerting.

III. The Product

A. Overview

The final Radius product can actively and accurately warn the user when it leaves a distance around their phone set at the user’s discretion. Upon leaving the perimeter, both the tag and phone will buzz to indicate a lost state. After a short period, the buzzing ends, but the user can push a button on the android application to activate a buzzer on the tag. Likewise, pushing a button on the tag will cause the user’s phone to ring. UWB technology provides high resolution tracking, with minimal deviation from actual distance measurement, even with high obstruction between tags (e.g. in a bag full of other items).

In both our PDR and MDR evaluations, our evaluators were not convinced that the low resolution of Bluetooth would be enough to create a reliable and consistent product, hence the switch to UWB. UWB provides a much higher resolution for distancing between items, and much more consistently than



simply using BLE. Incorporating UWB on a regular smartphone without it meant that an anchor tag needed to be placed near the phone for distancing. To do this, we incorporated an identical UWB module into a phone case for the smartphone, which would update distance readings to the Android app via BLE. As per the block diagram in Figure 6, the tag and anchor communicate distance data via UWB, and the Android application collects data from the tag via BLE.

Figure 5A: BLE Distance Measurements

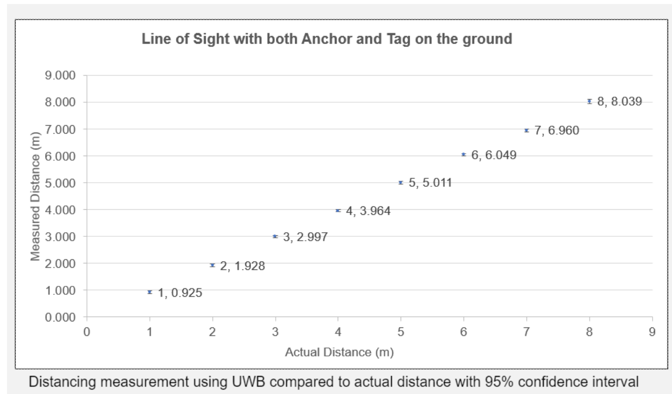


Figure 5B: UWB Distance Measurements - Unobstructed

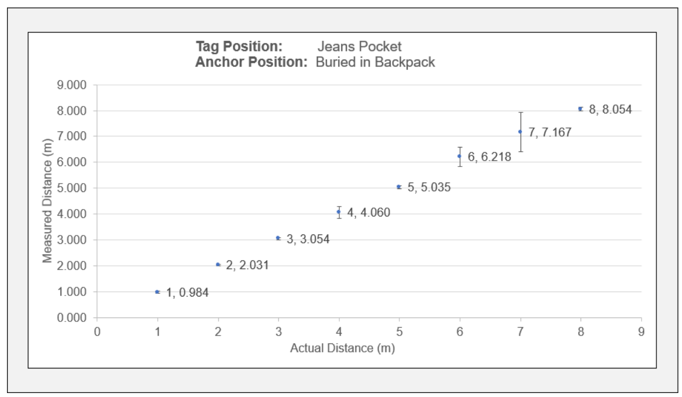


Figure 5C: UWB Distance Measurements - Obstructed

This new Radius product provides resolution of up to 18cm, and with a 95% confidence interval under 5cm. This is consistent up to 9m as in graph 2. Only using BLE already present in most smartphones, provides poor resolution. Figure 5A displays the large 95% confidence intervals associated with distance measurements using BLE RSSI. Comparing this to UWB distance measurements in Figure 5B, you can see that the 95% confidence intervals are much narrower with UWB. Additionally, the UWB system worked seamlessly when obstacles were present between the tag and anchor as seen in Figure 5C.

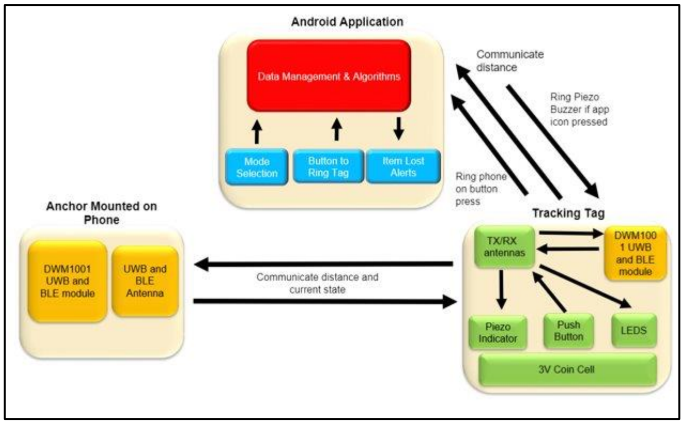


Figure 6: Final product block diagram

Figure 7: Final product Radius Tag

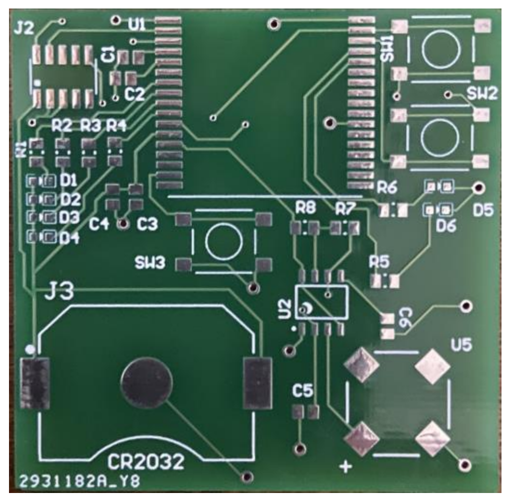
B. *Electronic Hardware Component*

In order to design the Radius PCB our team observed the DWM-1001 development board and made note of the components on that board that were necessary to implement the functionality of our product. From these observations and our product specifications, the following table was created to provide an organized reference list of the main components that were required for our custom PCB assembly.

Component Name	**Description**	**Function**	**Citation Reference**
DWM1001 RF Module	UWB/ Bluetooth Module	Transmit and Receive Bluetooth/UWB signals	[12]
CR2032 Coin Cell	Power Source	Portable, long lasting power source	[13]
Piezo Indicator	3V Buzzer	Alert user of missing item	[14]

After pinpointing the components necessary to implement our design, our team made various design choices that impacted the size of our device. In order to save space in our layout, the decision was made to use an external J-Link debugger to program the DWM-1001, as opposed to the on-board IC debugger used on our DWM-1001 development kit. This decision also allowed us to avoid the micro USB port and voltage regulation required to interface with the on-board IC debugger. Additionally, the use of the CR2032 coin cell provided portable power and sufficient battery life in a sleek package that did not exceed our size specifications.

Altium designer was the software our team utilized to design the PCB. The initial steps taken to begin the design process involved creating an Altium PCB library to store the footprints for all the components involved in our assembly. A schematic describing the physical connectivity was illustrated in Altium, and a prototype-implementation of the schematic was implemented on a breadboard where the functionality of the design was verified. The philosophy for our PCB construction was to use a two-layer board in which most of the routing was present in the top layer with minimal routing in the bottom layer. The bottom layer was filled with a copper pour and acted as a ground plane for the device. Having a two-layer board was important to the group in terms of minimizing manufacturing costs and staying under budget. The layout was done by hand to ensure as little routing in the bottom layer as possible because excessive routing in the bottom layer can disrupt the continuity of the ground plane. After layout was complete, a design rule check was carried out to ensure that all manufacturer specifications were met. After passing the design rule check, Gerber production files were generated and uploaded to jlcpcb.com for manufacturing. Below you can view an un-populated Radius custom PCB:



All SMT components were hand-soldered onto our custom PCB. A populated version of the PCB seen above can be viewed below:

After assembly, we successfully programmed our device using our external J-Link programmer. Then, our PCB was connected to a pre-existing UWB module that had been established for software testing previously. We found that our device was able to calculate distance accurately, establishing that our device was capable of basic functionality. We then proceeded to test the secondary components such as the indicator to verify that it produced audible noise when enabled. The only problem encountered in our design was short battery life from the CR2032 coin cell. After thorough analysis, we have determined that the CR2032 is in fact suitable as a source for our design but will not produce suitable battery life until our software is optimized for low power consumption and sleep mode.

C. Functionality

Almost all elements of our product were functional by CDR. We had a button in the app that could ring the tag. We had a button on the tag that could ring the phone. Both devices rang when they were more than 5 meters apart. We could not select between priority modes by CDR, but can do so now via the Android app.

D. Performance

Almost all specifications were met. Our final board had an area of 22.3 cm^2 and a height of exactly 5mm, just meeting the size specifications of 25 cm^2 area and 5 mm height. Audible sounds were produced in all 3 scenarios: leaving an item behind, ringing the phone from the tag, and ringing the tag from the phone. We had the capability of running a single priority mode by CDR. We can now switch between two modes as specified via the Android app. Our battery life specification was the hardest to meet. By CDR, our product could only last about an hour on a CR2025 coin cell. We used a much larger battery (670mAh) to ensure proper functionality throughout all of CDR. We now have a version of the product that can run for over two weeks on a CR2025 coin cell (170mAh) and can theoretically get this product to last 5 months with an efficient regulator.

IV. Project Management

Figure 2 illustrates our projected specification requirements for our final project. Our goals for MDR were accomplished, but minor alterations to our goals were made at the start of the project. Our first approach was to distribute responsibilities to each team member, but later decided on appointing leaders for each aspect of the project, allowing for overlapping efforts. Our main goal for MDR was to produce a successful prototype capable of handling transmissions from the tag to the phone, which was predominantly software based. Although we have made progress on the development board, next semester will be aimed towards hardware design. Specifications 4, 6, 7, 8, 9 from Figure 2 have been satisfied as required for MDR.

Figure 3 is a Gantt chart that displays the responsibilities for each team member in the first semester.

Refer to Appendix C to see examples of our team structure and approach to the project, as well as examples of leadership and initiative taken by each team member.

V. Conclusion

Our project currently consists of a functioning Android application that can approximate the proximity of paired devices. The Android application consists of a simple user interface that informs the user if an item is about to be left behind. Our team collaborated to determine the best way to approximate proximity from the RSSI (Received Signal Strength Indicator) signal and tried two different approaches for implementation. The zone-based tracking algorithm uses empirical measurements to categorize the RSSI signal into three different zones that represent an item being close, moving away from the user, and being completely lost. The second approach utilizes state-based tracking to determine proximity based on the observation of the history and trends of the RSSI signal that has been collected during program execution.

The next step in our development process will focus on two-way communication from the Android application to the Bluetooth module to provide interrupts for ringing the piezo buzzer on the tracking tag. We are aiming to have our localization technique finalized at the start of the Spring semester, then promptly begin PCB design.

Acknowledgment

Team 1 would like to acknowledge Professor Aksamija for providing wisdom and support both while selecting a project and during design and implementation phases. We would also like to thank Professor Hollot for organizing bench side meetings and providing feedback on our progress. Finally, we'd also like to thank Professor Burleson and Professor Kwon for evaluating the progress that has been made so far.

References

[1] J. Gleick, Faster: the acceleration of just about everything. London: Abacus, 2011.

[2] “U.S. Census Bureau QuickFacts: United States,” *Census Bureau QuickFacts.* [Online]. Available:

https://www.census.gov/quickfacts/fact/table/US/SEX255218. [Accessed: 05-Dec-2019].
[3] R. Balani, 'Energy Consumption Analysis for Bluetooth, Wi-Fi and Cellular Networks', Research Paper, University of California at Los Angeles, Los Angeles, n.d.
[4] P. Prasad, 'BLE vs Wi-Fi: Which is Better for IoT Product Development?', Cabot, 01-Feb-2018.
[5] A. Alarifi et al., 'Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances', Sensors (Basel), vol. 16, no. 5, May 2016.
[6] 'EFR32MG1 Mighty Gecko Multi-Protocol SoC Family Data Sheet'. Silicon Labs, 2016.URL: https://www.silabs.com/documents/public/data-sheets/efr32mg1-datasheet.pdf
[7] 'CPE-120 Datasheet'. CUI Inc, 07-Dec-2007. URL: https://www.cuidevices.com/product/resource/cpe-120.pdf
[8] "APDS-9007-020," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/broadcom-limited/APDS-9007-020/516-1719-2-ND/1785429. [Accessed: 23-Jan-2020].
[9] "Find Your Keys, Wallet & Phone with Tile's App and Bluetooth Tracker Device," *Tile*. [Online]. Available: https://www.thetileapp.com/ [Accessed: 23-Jan-2020].
[10] "Find Your Keys, Wallet or Anything You Don't Want to Lose," *Chipolo*. [Online]. Available: https://chipolo.net/en. [Accessed: 23-Jan-2020].
[11] 'Android Bluetooth Documentation'. Google. https://developer.android.com/reference/android/bluetooth/package-summary
[12] "DWM1001C Module," *Decawave*. [Online]. Available: https://www.decawave.com/product/dwm1001-module/. [Accessed: 09-May-2020].
[13] "CR2032," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/panasonic-bsg/CR2032/P189-ND/31939. [Accessed: 09-May-2020].
[14] "SMI-1027-TW-3V-R," *DigiKey*. [Online]. Available: https://www.digikey.com/product-detail/en/pui-audio-inc/SMI-1027-TW-3V-R/668-1662-1-ND/9083417. [Accessed: 09-May-2020].
[15] 'Bionic (software)', Wikipedia. Feb. 06, 2020, Accessed: May 09, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Bionic_(software)
[16] 'About eCos'. http://ecos.sourceware.org/about.html (accessed May 09, 2020).

Appendix

A. Design Alternatives

We considered Wi-Fi, Ultrawide Band (UWB), and RFID protocols as means to estimate proximity. We decided against using Wi-Fi because we discovered that it would result in a higher power consumption [3]. The tags would also need to be connected and registered to a Wi-Fi access point, which would be difficult to implement without a screen and text input. Also, the tag would become very large.

We also considered Ultrawideband technology which would offer superior ranging characteristics [5]. UWB is already used in factories and warehouses to perform inventory management and indoor localization [5]. However, the experience we are trying to achieve revolves around a user's smartphone. Our current solution needs an attachment to keep the anchor on the phone. We hope that

Similarly, using RFID would add bulk to the smartphone, and it would also not be a good choice because it is not designed to provide good localization data.

Bluetooth would be a great choice because Bluetooth radios are already integrated into smartphones. They are produced in much larger quantities which would help with cost. However, the inferior accuracy makes this method harder to trust when analyzing distance measurements. Combining the developed Bluetooth

B. Technical Standards

A lot of the software for our project is open source, but controlled by single entities, for example, the Java runtime and Android OS used in our app is managed by Google. Android is built on top of the Linux kernel for which there is a standardization effort under ISO/IEC 23360. The kernel shipped with Android devices is not compliant to the standard due to licensing reasons. [15] The firmware running on the tag is written in C which is standardized under ISO/IEC 9899:2011. This firmware is processed by the eCos real time operating system.[16]

Our hardware incorporates two standards in the IEEE 802.15 group, 802.15.1 for Bluetooth Low Energy, and 802.15.4-2011 for Ultrawideband. Both wireless standards are low power, so they fall under FCC 47 CFR 15 allowing unlicensed broadcasting in certain regions of spectrum.

C. Testing Methods

In addition to informal testing done each every day when tweaks are made to the code, we have developed an Android app with inputs for test duration, sample rate, and buffer size (used to average very noisy RSSI reading) to analyze the system's performance under various circumstances. All test run so far have been conducted to test the Android Application "Data Management and Algorithms" block. In playing with the buffer size input, we found that a buffer size of roughly 20 points generated a fairly clean output corresponding to distance if continuously accepting data points (sample rate = fastest = 1 sample per RSSI reading). However, if we decimated every other RSSI reading (sample rate = 2) and reduced the buffer size to 10, we would still update the Android application at the same frequency as with buffer size = 20, sample rate = 1, but the data was too noisy which is why we stuck with buffer size = 20 for MDR. After arriving at our ideal buffer size and sample rate, we ran the Android app at distances of 1m, 2m, 3m, and 4m in the MDR conference room and collected 1000 RSSI data points for each. We used this data to calculate standard deviations, variances and 95% confidence intervals at each distance. Using these values, we accurately set 3 ranges of RSSI values corresponding to relative distance of the tracking tag from the phone. However, these ranges were hard-coded for the MDR room environment. The results of this experiment can be seen in graph 3 below.

After switching to UWB for ranging our accuracy on distance estimates significantly increased. We conducted the test the same way as we did for BLE, but our data points were not RSSI measurements. Rather, the DWM1001 uses its internal location engine to calculate distance based on time of flight. Distances measurements became much more accurate as seen in Figure 5.

To test our power specification, we will run our system continuously for one day using a new battery. We will find out how much voltage is left on the coin cell after that day. By comparing this voltage to the 3dB voltage of 2.12V, we can estimate how long it will take to reach the 3dB voltage if our system were continuously running as it would be when in use.

We will test the specification for leaving the vicinity of your phone (or leaving vicinity of tag with phone in hand) by moving away at a walking pace. If our system can recognize that you are moving away with no obstructions, we can prove the simplest case works. However, to prove our system is robust, we will also conduct the same test when the phone is in your pocket to make sure the system works even with that high-level of interference.

All other specifications will be simple to test. Pressing an icon on our app will ring the piezo buzzer on the tag. A push button on the tag will ring the phone. The area and thickness specifications will be measured with calipers.

D. Team Organization

Our RADIUS team is composed of two Computer Engineering students: Ibrahim Tahoun and Titus Rwantare, and two Electrical Engineering students: Richard Brown and Cormac Kennedy. The project consists of two pairs, each composed of an electrical and computer student to handle different aspects of team responsibilities. On the Android Application end, Cormac and Titus are responsible for handling incoming signals to the phone and evaluating trends in signals. On the Thunder Sense development board end, Richard and Ibrahim are working to configure the board layout to connect with I/O components, power consumption, as well as transmitting signals to the phone. Our team is handling the project considerably well given our limited schedule overlap. During our meetings, we set short-term responsibilities for each team to cover over the next week. Our Gantt chart has a layout of our long-term objectives, but due to our team effort being focused on the application for MDR, all team members have taken initiative to learn and develop the Android application side of the project.

At the beginning of the project, our team struggled to communicate on a unified platform, until Cormac and Titus took initiative of setting up Slack for the team to use and linking it with a GitHub project. This enabled us to keep track of our responsibilities, share progress, and update our code all on one platform, increasing transparency between team members.

Cormac has helped the team by taking extra time to edit and organize the application code to be uniform so that our progress could remain consistent. Given that team members practice different coding habits, it was key to keep clarity among the team when working on the project at different times. Detailing the code to clearly to keep track of revision history was encouraged to cement habits that are used in the professional field.

At the start of the project, Titus helped guide the initial steps of the project by researching the requirements and benefits of Bluetooth over other mediums of communication, unifying team efforts in our approach to the project.

Richard is the leader for PCB design for the project. Although our efforts were not focused on hardware this semester, he took initiative to create a full library in Altium of the components that will be laid out in our final product, having foresight of what will likely be one of the hardest parts of this project. Ibrahim and Richard began working on the software to test the research that was collected at the start of the semester. Combining their shared electrical and programming knowledge to start the processing algorithm helped kickstart the application successfully.

Ibrahim is the team manager and has taken the responsibility of keeping track of the team’s progress throughout the semester. He created schedules for the team to follow when team members are working separately, organized weekly meetings with the advisor, and set up PDR and MDR evaluations.

Shared initiative and care for the project has allowed the team to progress persistently on the project, with negligible struggle in the form of teamwork. As the team faced difficulty on one half of the project in the first semester, they adjusted their responsibilities to overcome challenges produced in the Android application to produce a demo-worthy prototype.

E. Beyond the Classroom

Cormac Kennedy – Until this semester, I had no experience in wireless communication. Although engineering a Bluetooth product was new to me, my experience coding hardware in C allowed me to quickly read and understand the tracking board source code. In viewing this source code, I learned a decent amount about the GATT profiles used to enable Bluetooth communication. While running experiments to characterize RSSI values at different ranges, I learned precisely how much noise to expect in Bluetooth signals at these ranges. This knowledge along with noise-related knowledge gained while reading on Bluetooth alternatives have provided me with a good idea of advantages and disadvantages of various wireless mediums. When I graduate, I am going to work on PCBs installed inside planes and rockets where wireless communication is widely used.

Titus Rwantare – The most significant area I have developed in during this semester has been understanding the workings of the Bluetooth protocol. I found the Bluetooth SIG documentation to be incredibly detailed, but clear to read. I needed to learn how to develop in a new software suite for the Bluetooth Development Kit which introduced me to wireless radio terminology, but also wider used development protocols like SPI and I^2C. I also feel I’m more competent at using Version Control tools like GitHub and team communication platforms like Slack. These collaboration tools will forever be useful to my future endeavors as a professional.

Richard Brown – During the Fall semester I was able to refresh my Java skills and learn my way around the Android studio IDE. From this experience, I also learned about the software development life cycle, and how to develop software in teams by using tools such as GitHub for version control. The skills listed above even helped me successfully complete an interview for a Java Developer role with ISO New England in which I was extended an offer of full-time employment. Some useful resources that have helped me along the way are

YouTube, Stack Overflow, Silicon Labs reference guides, and my own group members.

Ibrahim Tahoun – My largest area of growth this semester has been understanding Bluetooth protocols. Learning the workings of attribute profiles to create Bluetooth entities and how to configure the development board to implement said attribute profile. Another thing I learned about was the Simplicity Studio SDK and Android Studio interface for programming both the application and the development board. Learning how to operate on new programming platforms and developing code as a team using tools like GitHub are skills that I am excited to carry into my career after graduation.