Final Report

1. Introduction

This paper will cover the audio communication and tracking solutions for the project assigned to Professor Johnstons’ lab 3 class. The goal of this project is for two teams, a base station team and a rover team, to locate 3 books hidden randomly throughout the Texas Tech University area known as Urbanovsky Park. Groups are tasked with not only tracking the rover team along their journey to find the books, but also to maintain constant and stable communication between the teams without using any existing communication infrastructure. This means that any use of Wi-Fi, GPS, LTE, Satnav etc. is prohibited. Once a book is found, the rover team must contact the base station team, request the code word location within the book, encrypt this code word and then securely send the data to the base station team to be decrypted. This paper covers the development process for an audio communication device and a transmitting beacon to accomplish two very important goals, the main method of contacting the other team and the tracking. The paper is divided into the fundamental engineering process steps which are the design stage, the modeling/simulation stage, followed by the manufacturing stage, and ending with the testing stage for both devices to be used in the project. A conclusion at the end of this paper will summarize the content as well.

1. Important Information

This paper will cover only the devices related to audio communication and signal transmission. I will mention the other components needed to fulfill this project in this section and then continue to the main focus of the essay. The main solution to communication without the use of pre-existing networks is to use radio modules that can transmit and receive signals on a specific frequency that is not regarded as Wi-Fi or Bluetooth or anything like that. So the initial research done on this project mainly focused on how radios worked, what kind of components were commonly used for radios and other devices that can transmit information and other various avenues of data concerning modules that did not rely on existing infrastructure. Research and information gathered about this project is found in the sources and references page of this paper. After a period of one week of research, our group crafted a critical design review, to summarize and concoct a plan of attack for the project that lay ahead. This is how we came up with our solutions. For example, our tracking solution is based on time of flight. This is a method for determining the distance between two signal sending devices by calculating the time it takes the signal to reach one of the devices, in our case, beacons. This calculation must be done on a device with a clock counter fast enough to count in the nano-second range to attain a resolution precise enough for the scope of this project. The other part of this project that is of great significance is the method in which we communicate the encrypted word between each team. Once a book has been found, we must make sure that if our signal happens to be intercepted by an opposing group, the word being transmitted does not get stolen. These two aspects of the project are key to our goal in addition to the methods for audio comms and beacons described in the rest of this paper.

1. Design

The designs made for this project must meet certain requirements derived from the project description. The devices used for audio communication need to be robust, long range, structurally sound, be capable of withstanding a variety of outdoor conditions, be capable of mild electromagnetic interference mitigation, and have an operating time of at most 3 hours. The devices used for location and tracking need to transmit a signal as well, and so have similar parameters, with a notable exception. The devices used for transmitting a signal periodically do not need any input from a user or team and must function autonomously, so the hardware is much simpler when compared to the communication device.

For the first step of design, a schematic for the Walkie-Talkie and the beacons were created. The microcontroller used for both the audio comms and beacon system is the Arduino Nano, chosen for its small size, wide range of application, and low power requirements. The Arduino Nano uses 5 V, so a 9 V battery regulated down is used to ensure proper power delivery [1]. In conjunction with the Nano on the audio communication device, an NRF24L01 Wireless transceiver is used for both transmission and reception. With a range between 1000 meters and 1500 meters, and an operating voltage of 3.3 V, this module fits squarely within our constraints and fulfills the range requirements for the comms system [2].

For the beacons, the Nano is paired with an EBYTE E32 900T20D LoRa Module radio transceiver module. The E32 has 7 pins needed for proper communication with the microcontroller. These pins are the mode pins M0 and M1, the “receive” and “transmit” pins RX and TX, the auxiliary pin AUX, and the power and ground pins, VCC and GND. For our application, we need the modes M0 and M1 to be pulled low, so configuring these pins in the software side of things or routing the pins directly to ground are both great options. The RX and TX pins are fairly self-explanatory, these pins are used for transmitting and receiving data between this E32 and the Arduino Nano. Now for what is perhaps the most interesting pin, the auxiliary pin. This pin is used to “indicate the working status of the module” [3]. This pin can actually alert the user as to whether or not the module correctly initialized all of its own parameters before use. This pin is crucial in our implementation because the Nano needs to properly recognize the model number and version of the LoRa module in order for the libraries to work. If this auxiliary pin is not functioning properly and the Nano fails to read the E32, then no data will be sent between the Nano and the E32. Another note is that the protocol for this module is UART or Universal Asynchronous Receiver-Transmitter. Rather than using SPI, which is a synchronized communication protocol, we chose to use UART, which is an asynchronous IC built into the circuitry of the module. This type of protocol allows for the transmission of serial data, which is necessary for sending a pulse of data during the project operation. A UART capable module was used because the only task required for these beacon devices is to transmit and receive simple text data and UART only needs 2 wires to accomplish this, so it is very simple to implement hardware wise [4]. Appropriate citation goes here x

This module can also operate at 5 V, which means that the same voltage regulator and battery used for the audio communication devices can be reused for this design. The device, however, operates on 3 V logic, meaning that the pins associated with data (i.e. not VCC or GND pins) can not be directly wired to the Arduino Nano, as this microcontroller utilized 5 V logic. We can get the desired logic level by one of two methods. One method would be to use a voltage divider circuit and calculate the relevant values to achieve a 3 V output. This works by having two resistors on either side of an output wire and depending on the values set can vary the output voltage given a specific input voltage. However, this is much more complicated and less effective option than the second one. Option 2 is to use a logic level shifter. A logic level shifter is able to take full advantage of a MOSFET to translate a high voltage or logic level to a lower one. To put it simply, the low side of the MOSFET circuitry is pulled high by a pull up resistor, in this particular case it is pulled up to 3.3 V, while the high size is pulled high to 5 V. Depending on what the gate of the MOSFET sees, it will alternate which side is pulled low and can thereby give us our desired voltage level. The logic level shifter used in this project is a bidirectional one, so data can be appropriately sent both ways to and from the Nano and LoRa.

The most attractive feature of this particular LoRa module is the range, as our initial goal to meet was a campus-encompassing 2 kilometers. The E32 900T20D boasts an optimistic 5.5-kilometer operating radius. An important note considered in the choice of this module is the conditions in which it would be operating versus the conditions in which this theoretical value of five and a half kilometers applies. Our project description lists an on-campus experience, and the university is riddled with buildings, trees, and other structures that could limit range and functionality of our modules.

In the unlikely event where our batteries decide to give out, the devices are designed such that the battery is quickly and easily replaceable, thereby fulfilling our battery design constraint. The previously mentioned Nano-friendly 5 V is achieved with a LM7805T Voltage Regulator. This same regulator is used for both the audio comms and beacon devices.

The walkie-talkie devices utilize a condenser microphone and a speaker to serve as the audio input and output, respectively. A general-purpose transistor was used in the condenser microphone circuitry due to an expected low amplitude pickup commonly associated with small condenser microphones in audio applications [5]. The 2N3904 Transistor uses 10K Ohm resistors in an amplification circuit as per the datasheet requirements during nominal voltage operations [6]. Polarized/Electrolytic capacitors are implemented between any components’ voltage in/output and their ground in order to help stabilize the voltage signal and minimize voltage spikes. A widely accepted and recommended value of 100 uF is used for this purpose [7]. The designs of the devices were made with EasyEDA and EagleCAD for a cleaner look and an easier conversion from schematic to PCB layout (only for the walkie-talkie).

For the walkie-talkie, the schematic was converted into a PCB using EagleCAD and a 3D model using Fusion360. The original schematic was drawn on EasyEDA, an online circuit development website. However, due to the unfamiliar user interface, tools, and settings for the schematic to PCB conversion, the circuit was transferred onto EagleCAD. Once the schematic was transferred, it was checked for errors with Eagles built in ERC, or Electronics Rules Check [8] which can catch any common errors, such as missing nets, improperly connected ports/pins etc. Then when the design was converted into the PCB layout, another check was performed, this time the DRC, the Design Rule Check to ensure that there were no outstanding manufacturing-specific constraint issues.

For the beacons, a PCB was not required, as all the components fit nicely on a proto-board and can be easily encased in specialized TTU project lab boxes with only minimal modifications for fitment. In practice, only 3 of these beacons are needed for full campus coverage, so the materials for building with these project boxes were more financially friendly choices as opposed to fully manufacturing specified PCBs for devices of this simplicity. Using the provided black boxes would also reduce our timeline, as we would not have to go through the trouble of what comes with 3-D printing an extra 3 containers, which was projected to take roughly 25 hours. Extra words just in case I do not meet 5000

As seen in Figure 1, the PCB trace layout is done using the Eagle auto-routing tool that determines the best course that each trace should take to get to its corresponding component and lays at 6mil in trace width. This is important because the trace width must be expanded for proper soldering (up to 50mil). The traces need only be slightly adjusted for the same reason.

Also included in the design illustrated in Figure 1 is a ground layer. This allows for any component that may need to be grounded to be connected directly to this layer and therefore does not require a trace, making the design simpler and the manufacturing process cheaper. This layer can also provide heat regulation or what is known as “thermal relief” [9] when soldering components on the printable circuit board and while in use, acting as a heat sink.

Map

Description automatically generated with medium confidence

Figure : PCB Layout designed prior to transfer onto 3d PCB model. Blue region is a ground layer to act as a heat sink, de-coupling capacitors are placed close to their respective components for steady voltage input/output

A 3-D model was also created with the use of Fusion360. Similarly, the conversion from schematic to PCB follows the conversion from PCB to 3-D model and utilizes a special tool built into EagleCAD [10].

The PCB is designed to fit in an average human hand [9], with a dimension set of roughly 4 inches wide and 6 inches long. This provides enough space for the solderable parts, which include the resistors, capacitors, the transistor, the speaker, and the chip boards we will be using, as well as allows for a casing to be built around the board, making the device more ergonomic and usable [11].

A picture containing circuit, electronics

Description automatically generated

Figure : 3-D Model of PCB. All the components associated with the walkie talkie will be on this PCB. The model includes holes at each corner of the board for mounting into an outer casing for protection and structural support, as required by the design specifications found at the inception of the project.

1. Model/Simulation

As previously stated, the beacons did not need any models to be 3-D printed nor any PCB’s to be manufactured, however, the circuitry was built and “simulated” on a breadboard before transferring to a more permanent solderable board. A 3-D model for the Walkie-Talkie encasing was designed in a CAD software. The software was the same that was used for the PCB design, so integrating the PCB into the encasing design was a smooth process.

Diagram, schematic

Description automatically generated

Figure : Schematic for the beacon devices. The Arduino Nano has a logic level incompatibility with the LoRa transceiver module and so the two modules are split by a logic level shifter to account for this discrepancy. Both modules are seeing power from the voltage regulator which is fed by our 9 V battery.

Once the dimensions and components of the board were set, some sub-design challenges were identified. There are a few pre-requisites that this enclosure needs to fulfill. Firstly, the case needs to be removable so that access to the inner components is quick and easy. This is so any issues that we may encounter with this PCB or the enclosure itself can be quickly and easily resolved by simply opening up the enclosure and re-attaching the top and bottom once we are finished with repairs or modifications. Secondly, the enclosure needs to securely hold the PCB in place within it to prevent any damage or rattling while in use. Lastly, the dimensions of the case need to fit the PCB snuggly so in the event of a drop, the parts will remain secure.

The first design challenge is achieved through a set of 4 stands that interlock on the bottom and top halves of the enclosure (See Figure 3). This is a simpler design and is more forgiving to print with a standard 3-D printing nozzle than that of other interlocking methods [12].

A picture containing electronics

Description automatically generated

Figure : Inside of case, focusing on the bottom half the walkie-talkie. The PCB will lie 3 mm above the floor of the case and sit below the 20 mm wall height, leaving enough room for the top half to snap into place.

The second design challenge is accomplished with the joining stands of the case that were previously mentioned. The PCB is designed with 4 drill holes on each corner at 9 mm from either edge. The stands that interlock the casing together will also pass through these drill holes and secure the PCB in place so that no rattling or movement will occur while the device is in use (See Figure 4). A 1/8 inch or 3.175 mm diameter was chosen as that is a standard drill bit size [13].

Graphical user interface

Description automatically generated with medium confidence

Figure : Inside of the case, focusing on the upper half of the walkie-talkie. Notice the interlocking receiver holes attached to the upper layer, as these will join to snap the encasing together.

The dimensions included the length, width, and height of not only the PCB itself but also the components onboard. The enclosure needed to be tall enough to allow for our tallest component, namely the transceiver (with antenna) which sat high inside the case [14]. So, the overall height of the case is set at 40 mm which will fit all components inside.

Diagram

Description automatically generated

Figure : 3-D case illustrating the dimensions for the model. A cutout for the speaker is included and shown in this model, which allows for audio to exit the case.

1. Manufacture

The beacons’ schematic was built on a breadboard to first test the device itself. Once the components were verified to be working as intended, they were all soldered onto a prototype board for a more permanent build. This board is then placed into a project lab black box, which comes prefabricated, made from plastic and with top and bottom removable halves similar to my custom walkie-talkie design. As can be seen in figure 7, some minor physical adjustments were made to the box in order to better fulfill our needs for this project, as these boxes all come in a standard size and shape. The boxes initially did not have an opening for our LoRa module to fit in, so a hole was made to accommodate this component. The sharp-eyed reader may notice that the LoRa module and the logic level shifter are actually on the same rails on the PCB, meaning that the high side, low side, and data side of the shifter and module are actually all receiving power/signal. This would obviously ruin the module, as the AUX, RX, and TX pins are not able to handle 5 volts. A specific cut was made on the board rails to electrically isolate the logic level shifter from the LoRa module and ensure proper functionality.



Figure : The project black box with our beacon circuitry on the solderable protype board within. This image illustrates how the box can be opened and closed for access to the inside as well as how the board itself is attached to the box.

The walkie-talkies were built on breadboards and a TinkerCAD illustration was made to help with understanding the wiring of the breadboard. When wiring the circuits on the breadboard, the schematic designed in the initial design stage is used as reference. Two standard, solderless breadboards are built in this stage. After thorough testing, all 3-D model files were sent to JLCPCB.com, a PCB manufacturing company, to complete the manufacturing stage of this device.

Since small fluctuations in voltage can interfere with components when working with audio circuitry, having multiple de-coupling capacitors is an important part of the build [15]. A de-bouncing capacitor of 0.1 µF is used with a button so that the button will have consistent operation. The Arduino Nano can be powered with 5 V and also has the ability to output 3.3 V to power any external components needed [16]. In our case, this component is the transceiver module used for communication between the pair of walkie-talkies. The output of the condenser microphone is connected to an analog port on the Nano, while the connections of the speaker are connected to digital ports. This is because we need to take a variety of voltage signals from the microphone, as these correspond to frequencies of your voice [17]. The speaker must simply be on or off depending on whether or not the voice signal is coming into the Nano, which is determined by the button. When using the NRF24L01, it is recommended to keep the actual module protected from electromagnetic interference, as this device is very sensitive to this particular type of signal weakening. Our solution was to simply isolate the component with aluminum foil so that any interference would be kept at bay and not alter our data that is being sent to the Nano. Another advantage of using an Arduino Nano for this particular audio application is that the audio signal being inputted to the Nano is able to be converted from the analog signal to the digital signal being seen by the speaker. The ADC (Analog to Digital Converter) can “sample the signal, then quantify it to determine the resolution of the signal, and finally set binary values and send it to the system to read the digital signal” [18].

Diagram

Description automatically generated

Figure : TKinter illustration. The circuits are powered by 9 V batteries connected to the top rail, which then goes through a 100 µF capacitor in order to stabilize the voltage [14]. After this capacitor, a wire connects the power to a 5 V regulator to power the Arduino Nano, the button, and the microphone. This power is also coupled with a capacitor of the same value and for the same purpose.

1. Testing

For the testing stage, two beacons were used on the field to gather information on real life range, signal strength, and data loss. A quick reminder that our goal for these devices is for a range of at least 1 kilometer with decent signal strength and minimal to no information loss when transmitting. One of the devices was situated in a parking lot south of Urbanovsky Park while the other beacon moved around predetermined locations in the area. A figure is included to better visualize the testing location. The testing of this area was all around good.

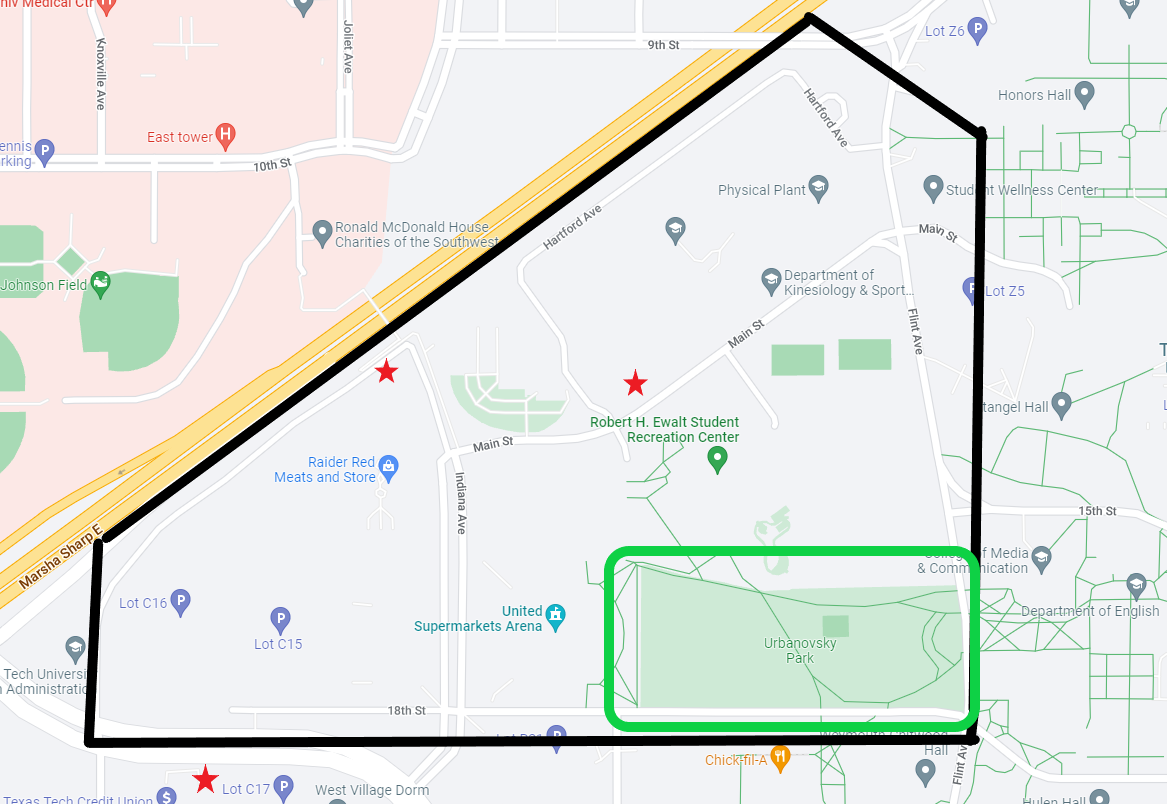


Figure : Shows a helpful map to better understand the testing method used for the beacons. Red stars are signal drop-off areas, where the signal being transmitted was never received or partially received.

The black outline is the testing zone and the data being transmitted from the static beacon was received in the entire inside area of the polygon. These red star drop off zones would typically be a concerning testing result and more modifications may needed to have been made to avoid losing precious data, however since the area of the project scope was reduced to the park land within the polygon, our team decided no further action was needed. A very valuable piece of data acquired during testing of these devices is that the transmitted signal was more likely to be missing or incomplete if the subject was moving at anywhere above 3 mph. This means that the recipient must be standing still in order to receive the intended message in its entirety. This is something that our team did have to take into account and altered our plan of action for demonstrating our project in operation. The distance from the two most opposite points on this black polygon is approximately 1.3 kilometers, which falls precisely in our 1 km to 2 km range yet falls short of the anticipated theoretical 5.5 km value given by the E32 900T20D datasheet. Our group was not discouraged by this lack of performance and continued with this module regardless.

The walkie-talkies were powered on, coded, and tested. Issues arose, mainly concerning the software, that prevented full voice-to-voice communication between the devices and thus were not tested to the extent I wanted. Let us quickly cover the problem, possible suspects, and a sub-conclusion for the testing stage of this project.

When powered on and programmed, the walkie-talkies are passively on receiver mode and able to output sound, however, when the button is pressed and puts the device into transmission mode, no voice can be heard out of the other device when spoken into.

I use an NRF24LO1 + PA + LNA transceiver in order to fulfill this function. For the walkie-talkies to work, it must be coded with the RF24 and RF24Audio libraries that contain functions for the NRF module [20]. These libraries are very large and extensive, and serve to initialize parts of the Arduino Nano and NRF module like the data rate, the SPI communication, the addresses etc. The various options available for setting allows the two transceivers to talk only to each other by using addresses and channel sets. An important note is that the NRF24LO1 + PA + LNA module used differs from the normal NRF24L01 transceiver in that it has an IRQ (Interrupt) which is “active low and only used if interrupt is required” [21]. This will be used for the final product and final demo, however for this testing stage, the interrupt pin is not used and only 7 out of the 8 pins are needed for text transmission (testing purposes only).

Knowing that the speakers are still able to transmit audio leads to a few possible issues. The first of these possibilities is that the microphone or microphone circuitry is not working. The eclectic microphone being used has a low output voltage and this needs a preamplifier [22]. The circuitry may be faulty, causing the microphone to have either a really low output or no output signal at all. Another possibility is the transceivers are not properly initialized in the code. This would cause a mix up in the functions of the pins for the NRF modules and thus data would not be interpreted correctly [23]. There are many important set ups that must be done in order to properly utilize the libraires to run these devices and my research has not been extensive.

Another test was conducted using sample code intended for sending only text, which tells us if the transceiver is properly sending and receiving data. This, as seen in figure 7, reveals a telling clue that may suggest the issue lies in the code rather than the hardware. This test shows that the transceivers are communicating data, although the data is unreadable.

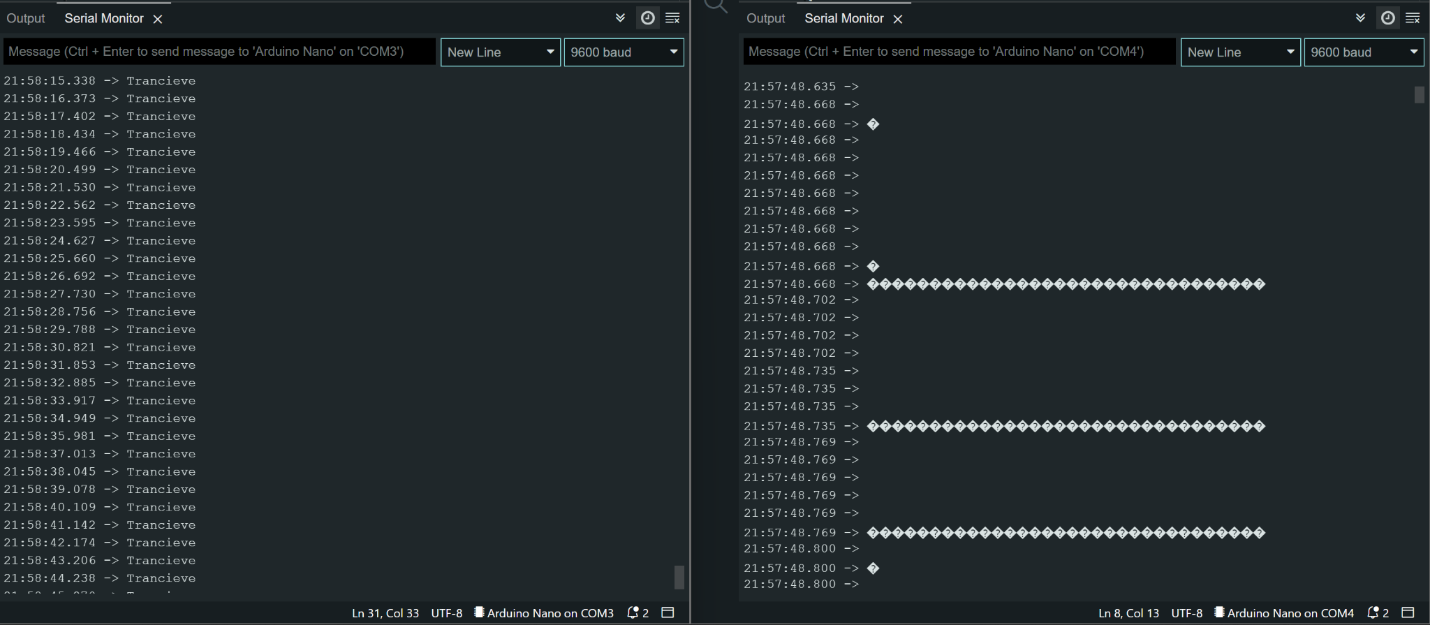


Figure : Serial monitor data from onboard the Arduino Nano while testing without using the IRQ (interrupt) pin on the transceivers. This test is to verify the proper transmission and reception of a simple "Transceive" message across two NRF modules. The data being read is illegible and distributed across time based on signal strength.

Figure 7 shows a baud rate of 9600 bps, which is recommended for this type of communication and was chosen for testing at very close ranges. This helps to visualize the communication issues.

The walkie-talkie devices are not communicating properly. The problem is likely in the assignment and initialization part of the code. After reviewing the circuitry on the breadboard and confirming that it is all in working order, the code used to transmit and receive will be re-written with extra attentiveness in the configuration portion of the software. The devices will then be tested similarly as before.

1. Conclusion

A few steps are taken in order to fulfill the project requirements mentioned in the introduction to this paper. First, in the design stage, we identify and note important specifications and constraints that the devices need to meet. These are range, battery life, protection from the elements, portability, and ease of maintenance. Also, within this stage, the components used in the devices are chosen with these standards in mind. For the audio communication devices, we used the Arduino Nano, the NRF24L01 transceiver, a voltage regulator, a preamplification circuit, and a 9 V battery [24]. For the beacons, the components were the Arduino Nano, E32 900T20D LoRa transceiver, generic bidirectional logic level shifters, a voltage regulator, and 9 V battery. All of these components were carefully selected to fulfill the project demands. For the beacons, a schematic was designed and tested on a breadboard before being moved onto the prototype board during our third stage. Luckily, no PCB or 3-D encasing needed to be designed or ordered for these devices, as there were better alternatives in the form of project lab boxes. After the beacons were built, they were taken to the field for testing, where our team found valuable information to be later used during project operation. For the walkie-talkies, the schematic is followed by a PCB layout using CAD software, which lays the foundation for the second stage, the simulation/modeling stage [25]. In this stage, a 3-D PCB model and casing is made. The casing follows the size requirements identified in the earlier stages. Third, in the manufacturing stage, the schematic is transferred onto another type of CAD software to aid in building two prototypes on two breadboards. During this stage the components are wired on the solderless boards and tested with a voltmeter to ensure proper connections. Finally, in the testing stage, the two circuits have code uploaded via the Arduino IDE and tested for functionality. These devices were non-functional and may need a Digital to Analog Converter to properly operate.

The purpose of this project was to implement a method of communicating via a team-built infrastructure and not relying on the pre-existing methods like Wi-Fi and Bluetooth and the like. While audio communication and beacon transmitting devices are important, they are only a piece of the overall project puzzle. Not mentioned in this paper are two other important components to this project done by my project lab mates. These are the written communication and encryption, and the tracking hardware needed to attain time-of-flight values. My teammates worked diligently to make sure that the two largest portions of the project were up and running and were extremely patient with my failure in the walkie-talkie endeavor. I feel it is necessary to mention this in the conclusion of this paper, as it was only briefly discussed in the body. I am thankful to my teammates and to Mr. Johnston for working with us to complete this project.

Sources/References

1. <https://www.circuitstoday.com/arduino-nano-tutorial-pinout-schematics>
2. <https://components101.com/wireless/nrf24l01-pinout-features-datasheet>
3. <https://www.electronicshub.org/bidirectional-logic-level-converter/>
4. <https://www.circuitbasics.com/basics-uart-communication/>
5. <https://www.neumann.com/homestudio/en/what-is-a-condenser-microphone>
6. <https://components101.com/wireless/nrf24l01-pinout-features-datasheet>
7. <https://www.electronics-tutorials.ws/capacitor/cap_2.html#:~:text=Typical%20values%20of%20capacitance%20for,smoothing%20capacitors%20in%20power%20supplies>
8. <https://www.autodesk.com/products/eagle/blog/design-rule-check-pcb-layout-basics-3/>
9. <https://absolutepcbassembly.com/what-is-pcb-ground-plane/#:~:text=It%20is%20when%20a%20clear,several%20parts%20of%20the%20PCB>.
10. <https://www.autodesk.com/products/eagle/blog/fusion-360-integration-eagle/>
11. <https://www.healthline.com/health/average-hand-size#:~:text=The%20average%20length%20of%20an,to%20hand%20size%20than%20length>. Make sure to delete this after the paper is done
12. <https://ergo-plus.com/fundamental-ergonomic-principles/>
13. <https://www.3djake.com/info/guide/3d-printer-nozzle-guide#:~:text=0.4%20mm%20nozzle%20as%20standard,resolution%2C%20precision%20and%20printing%20speed>.
14. <https://www.hubs.com/knowledge-base/standard-drill-bit-sizes-cnc-machining-conversion-tables/> this is also a valid source to use in this paper
15. <https://www.addicore.com/NRF24L01-PA-LNA-p/ad269.htm>
16. <https://www.protoexpress.com/blog/decoupling-capacitor-use/>
17. <https://passive-components.eu/capacitor-selection-for-coupling-and-decoupling-applications/#:~:text=Decoupling%20capacitors%20are%20used%20in%20electronic%20circuits%20as%20energy%20reservoirs,reliability%20of%20an%20electronic%20circuit>.
18. <https://diyi0t.com/arduino-nano-tutorial/>
19. <https://www.arrow.com/en/research-and-events/articles/analog-vs-digital-what-is-an-analog-signal>
20. <https://www.arrow.com/en/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters#:~:text=ADCs%20follow%20a%20sequence%20when,its%20sampling%20rate%20and%20resolution>.
21. <https://www.tinkercad.com/dashboard> this is a valid source of well being
22. <https://www.arduino.cc/reference/en/libraries/rf24/>
23. <https://components101.com/wireless/nrf24l01-pinout-features-datasheet>
24. <https://www.nutsvolts.com/magazine/article/how-to-design-a-microphone-amplifier>
25. <https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/>
26. <https://www.javatpoint.com/arduino-serial-serial-begin>