PDF-XChange Editor
Click to BUY NOW!
www.tracker-software.com

Low-Cost Arduino-Based Radio Frequency Power Monitoring Sensor Network (Final Report)

[redacted]

Activity Report

Abstract—As 5G telecommunications equipment becomes more prevalent, a lack of public understanding of the long-term health effects leads to uncertainty and fear. Research into the long-term health effects is minimal, and most of what has been done is focused in Europe, rather than the United States. The research into these effects and on radio frequency (RF) is largely performed in controlled conditions in a laboratory, which does not accurately reflect the modern RF landscape. Our project is a low-cost sensor network which monitors the signal strength of a specific telecom band in order to better understand how it behaves over the course of a long period of time. Our design utilizes LoRa radio protocols in order to create a distributed network of custom sensors tuned for 2.1GHz frequencies. This allows us to accurately plot the signal strength over the course of several days with minimal data loss.

Index Terms—electromagnetic field monitoring, 5G communications, LoRa, low-cost

1 The Problem

Research into radio frequency (RF) and its effects is mostly based in European laboratories. European research is beginning to implement environmental measurements to create more realistic exposure maps, however United States research remains primarily lab based. Our system is a low-cost, Arduino-based monitoring system which allows us to measure environmental RF signals around the University at Albany campus. This system utilizes long-range communication protocols (LoRa) to transmit data about RF field strength to a Raspberry Pi central hub. This allows us to create a network which can be used to research RF trends and possibly inform public health studies and officials.

[redacted]

2 Inspiration

Our team primarily used two published papers as inspiration and to establish a starting point for our designs. The primary work we used was research performed in France, in which an Arduino-based measuring system was designed [1]. This paper focused on a different frequency band than we plan to examine, and they had a very high noise floor which resulted in most of their findings being processed as noise. This is a problem we aim to address by focusing on a different frequency band, as well as designing a more stable circuit. We also referenced a Sensors publication which attempted to establish a method to measure signals using different low-cost software defined radios (SDR) [2]. SDRs, however, tend to be expensive, even on the cheaper end of the spectrum, as well as requiring special software to capture and process data making it difficult for mass use. Our system focuses on being easy to set up and implement, as well as being of comparable accuracy for an even lower cost.

PDF-XChange Editor
Click to BUY NOW!
www.tracker-software.com

Design Constraints

Our design has several critical design constraints we must factor in:

- **Impedance Matching:** When working with signals at particularly high frequencies, the impedance between components must be matched to prevent the signal reflecting back and ensuring maximum power transfer. This can be accomplished by using proper widths of PCB traces as well as using a matching network of capacitors, resistors, and inductors.
- **Component Power Restrictions:** The Arduino board we are using is the Arduino MKR WAN 1310 which has a very strict operating voltage of 3.3V to the Analog and Digital I/O pins. This means that all components and devices connected to this board must be able to operate at this voltage.
- **Low Cost:** Our design must be a low-cost solution to RF measurement. Commercial SDR technology was ruled out due to cost as most are greater than $200. We require our system to come in below this figure for viability. Otherwise we would use a commercial SDR for data collection.

4 Proposed Solution

Our design is based on an Arduino MKR WAN 1310 LoRa-enabled microcontroller (referred to as "Arduino" for the remainder of this paper), and a custom designed PCB shield which houses the circuit to capture and convert the signal to a usable power density value. This shield consists of an antenna connection to capture signals, which are then passed through a series of filters and a commercial RF detection component before being passed into the microcontroller for processing and transmission. The Arduino then sends the information via LoRa to a Raspberry Pi which performs data processing and quality control. Figure 1 shows the input and output of our system in its final form.

Figure 2 shows the frequency bands used by the cell towers in our area from cellmapper.net. From this map, we found that the most used RF bands in our area are n66 and n71. Band n66 is a 4G LTE signal in the 2.1 GHz range and band n71 is a 5G signal in the 600 MHz range. We used a Keysight FieldFox RF Signal Analyzer to determine which band to focus on. Given the relative strengths of the two signals, we chose to focus on the n66 band as it was the more powerful signal in our area of interest.

Once we determined which band to focus on, we could install our band-pass filter circuit with the RF detector on our custom PCB, and use the Arduino to measure the power levels. The Arduino then uses a LoRa transmitter to send the power level data to a central hub which will process the data and be able to generate a CSV file of long-term data, as well as displaying a short-term live plot of the incoming data stream.

4.1 Engineering Diagram

Figure 3 shows a block diagram outlining our solution and the flow of information through components. The dotted magenta block shows the components on our PCB shield which will have a physical connection to our Arduino. The green block represents the data network.

4.2 System Requirements

4.2.1 System Users

Our design is intended for use by anyone. Our system should be able to be set up and connected to a network without extensive user input. Once our device is deployed it will be able to collect information as long as it has power, and it will transmit collected data on a regular schedule. That data will then be processed on the Raspberry Pi and a user should be able to view that information without requiring extensive programming knowledge.

4.2.2 Functional Requirements

Our design has three main functional requirements:

1) **Wireless Data Transmission:** Our system requires wireless data transmission using the LoRa to collect data from the monitoring stations back to a central hub. This allows us to create a network which can cover a large area and still transmit

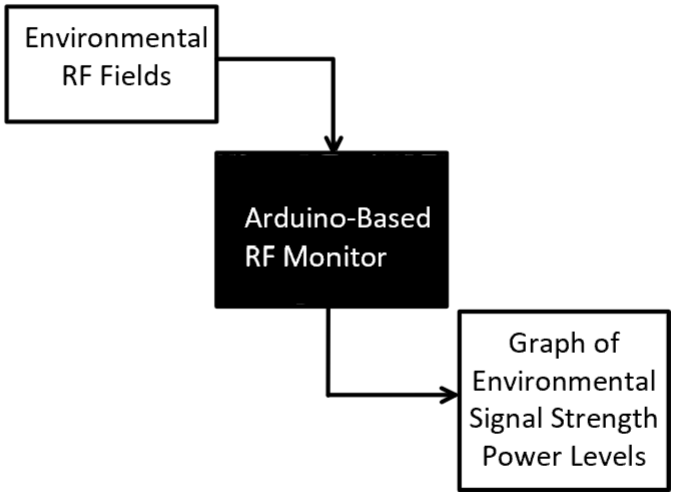

Figure 1. System inputs and outputs.

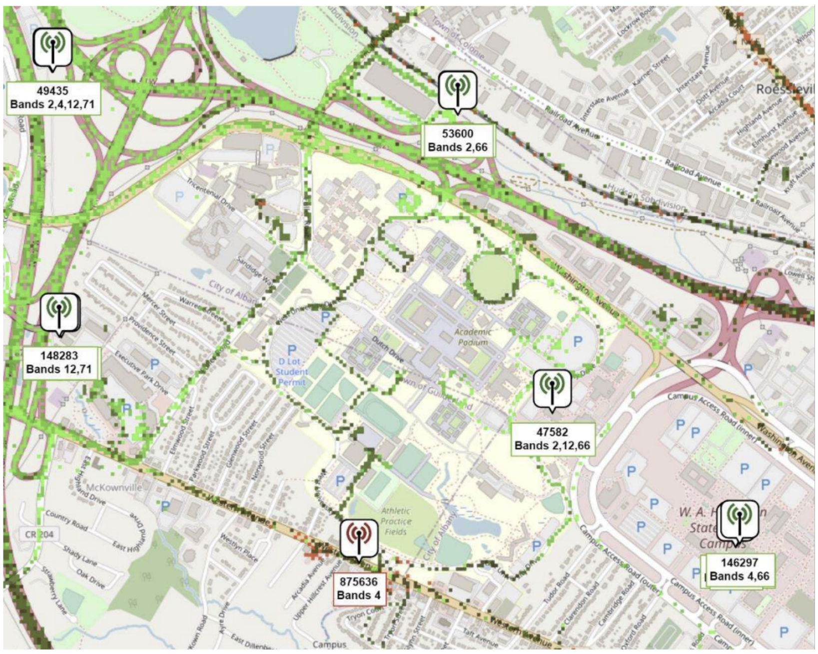

Figure 2. Map of the cell towers around the UAlbany campus and the frequency bands used.

information without requiring a Wi-Fa connection.

2) **Convert RF Signal to DC Voltage:** Our system must be able to convert RF power levels to a DC analog voltage for easier measuring. This will allow us to use an inexpensive ADC to capture these power levels for further processing.
3) **Receive a Wireless Signal for Processing:** Our system must be able to take in RF signal from a collection antenna and measure the power level of the signal.
4) **Cloud Computing Connectivity** Out design must send all data collected to a cloud service for further data processing and storage. This will make our data easier to view and manage, as well as making it easier to analyze the data with machine learning tools.
5) **Machine Learning for Data Processing:** Out system will utilize machine learning tools to analyze data sets and establish an RF exposure heatmap.
6) **Accuracy:** Our design requires that our

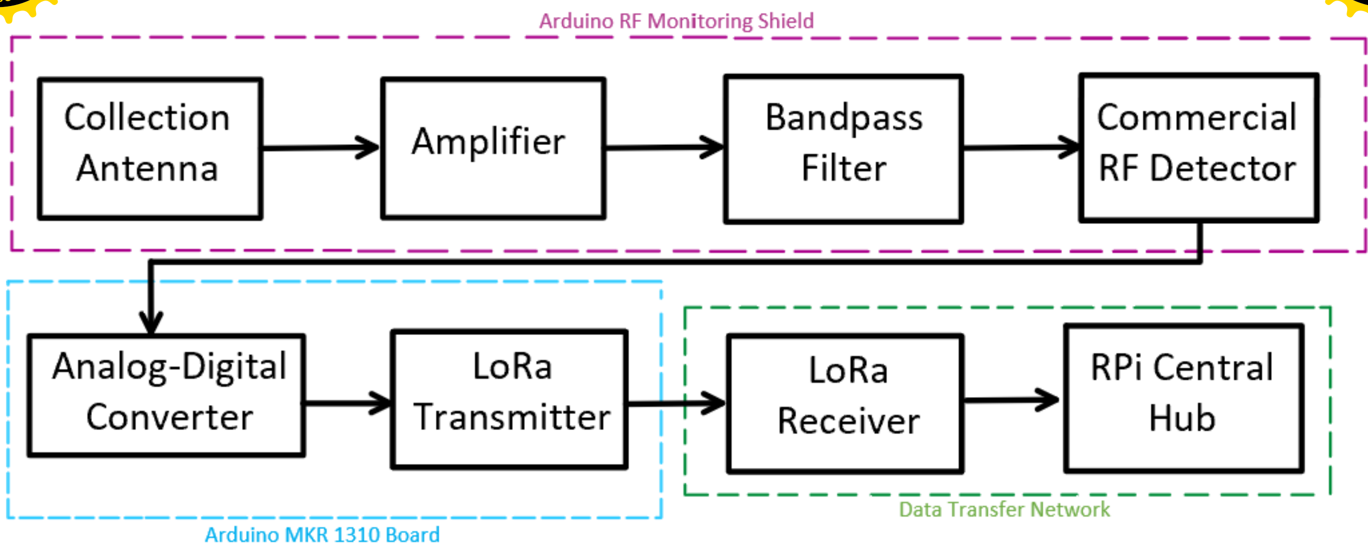

Figure 3. Block diagram outlining our solution and the flow of information through components.

system be able to capture RF signal data with an accuracy comparable to low level commercial SDR technology. Our current benchmark is the ADALM Pluto module which will also be compared to a high-quality RF Signal Analyzer.

4.2.3 Non-Functional Requirements

As well as the functional requirements listed above, our design also has a number of critical non-functional requirements:

1) **Scalability:** Our design should be easy to scale to increase the size of our network. This would allow us to cover both larger areas and achieve a finer resolution when generating a heat map of the covered area.
2) **Size:** Our final design should be small enough to be placed anywhere. Our system should be able to be placed both indoors and outdoors and be unobtrusive in order to capture data as naturally as possible.

4.3 Justification

We chose to approach the problem using the Arduino and custom shield for two main reasons. It allows us to customize and fine tune the noise floor and accuracy of our system. It is also a much lower cost solution than using commercial SDR technology.

4.4 Ethical Concerns

Our system has very few ethical concerns that we are aware of. Because our system requires very little user interaction it has very little chance of being misused. Once potential misuse concern would be the reconfiguration of our system to operate on frequencies that we are not legally allowed to operate on. This is a minor concern as it would require a significant redesign of our shield. There are possibilities of liability concerns if a user were to use our data and receive a potentially inaccurate measure of their exposure. However, since our intended use is for research purposes and will be tested against commercial technology for accuracy this is a very minor concern. Our system also does not present any user data privacy risks since our system is only capable of measuring the power and has no way of reading any data being sent.

5 Design Goals

5.1 Semester 1 Goals

By the end of the Fall 2023 semester, we had created our first functional prototype PCB test board. Our focus during the fall was researching how to interact with high frequency signals and identifying the signals to focus on for our project. We had also begun work on cloud computing integration, the anti-aliasing filter,

...d preliminary testing to prove our design concept was feasable.

5.2 Semester 2 Goals

By the end of the 2023-24 Academic year we were able to create three enclosures capable of being deployed and capturing data for extended periods of time. We had successfully created a LoRa radio network which allowed our devices to transmit the collected data to our raspberry pi with minimal data loss or corruption. We had also developed a weather-resistant, 3D printed enclosure for our circuitry which allowed it to be placed outside to better capture signals. Our system could handle doing most of the key data processing on the raspberry pi and displaying a live graph of the incoming power data.

6 Physical System Design

6.1 System Architecture

Our system is comprised of a custom RF monitoring PCB, an Arduino MKR 1310, and the data transfer network. Each of these components contains subsystems that work in conjunction to take in environmental RF fields, and output a CSV file containing the power levels from each sensor.

6.2 System Components or Subsystems

6.2.1 RF Monitoring Arduino Shield

The RF monitoring Arduino shield is used to convert the high frequency RF signal to a low frequency analog voltage which correlates to the power density of the signal. Our RF monitors will only be focused on the n66 band. This band was chosen due to its higher relative power compared to the n71 5G band. The layout of this PCB can be seen in Figure 6.2.1.

A four layer board stackup was used to help isolate RF signals and reduce interference. The stackup for the PCB consists of the RF signals and a ground plane on top two layers, and a power plane and other signals on the bottom two layers. In addition, a trace width calculator was used to determine the trace width needed for 50 Ohm characteristic impedance along

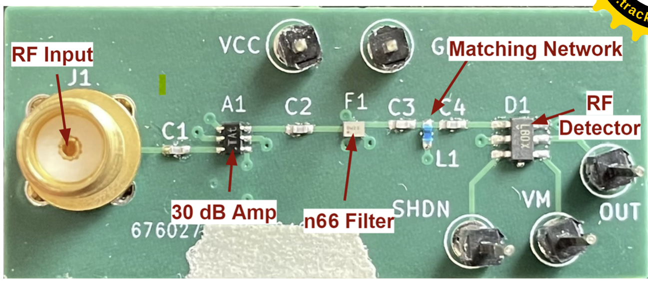

Figure 4. Layout of RF monitoring circuit test PCB for 2.1GHz

the RF lines. This allows for maximum power transfer of the signal from the antenna to the RF detector chip. The circuit consist of a vertical coaxial SMA connector which an RF antenna is connected to, as well as a 30 dB amplifier and passive bandpass filter for the n66 band that is being measured. After the bandpass filter, the LTC 5530 RF detector chip is used to convert the RF signal to an analog voltage related to power density. However, we found that the RF detector we are using does not have a 50 Ohm input impedance, and rather the input impedance varies with the input signal frequency. This means that an impedance matching network was needed to convert the 50 Ohm output of the bandpass filter to the input impedance of the RF detector chip. All of these components are labelled in Figure 2. For this test board, we connected jumpers wires to the outputs of the RF detector chip which are then connected directly to the Arduino MKR 1310 for testing and data collection.

6.2.2 Arduino MKR 1310

The Arduino MKR 1310 is a specialized Arduino board for LoRa radio work. It is an open-source board with the ability to connect to different pre-existing LoRa networks and devices, as well as allowing us to create a new, custom network. The board has a fairly strict operating voltage of 3.3V to its analog and digital pins. We are using this board to perform some preprocessing and averaging of collected data before utilizing its built-in LoRa capabilities to send the captured data on to our Raspberry Pi. The layout of this Arduino can be seen in Figure 5.

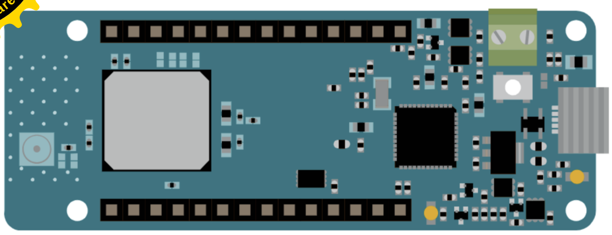

Figure 5. Arduino MKR 1310 layout

6.2.3 Data Transfer Network

Our planned device data transfer network consisted of three Arduino MKR 1310 boards each with their own RF monitoring shield, a Raspberry Pi Model 3 B+ central collection hub (central hub) and the Microsoft Azure Cloud Computing Service. The layout of this network can be seen in Figure 6. The collected data is preprocessed by the Arduino board and shield and is then sent over LoRa radio transmission to our central hub. The data is sent and collected multiple times each minute, and an average value of the data collected from each node is taken. This data is then stored on the Pi as a CSV file which can be processed by a user. We also created a live display of the data for a limited window, 2-3 hours, to be able to visualize short-term trends in the data. Azure was not able to be implemented due to several complicating factors.

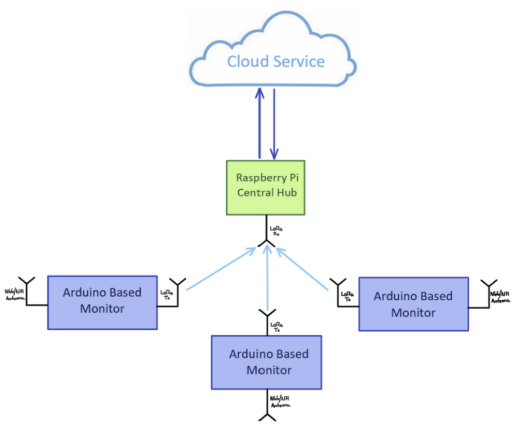

Figure 6. Diagram of LoRa and Microsoft Azure cloud network

6.3 Engineering Standards

6.3.1 LoRa

LoRa (short for “Long Range”) is a radio communication technique. It uses sub-gigahertz radio frequency bands and consumes very little power while having the ability to send data over fairly long ranges with a small antenna. LoR (Wide Area Network) is the communication protocol along with its associated system architecture. It is an official International Telecommunication Union (ITU) ITU-T Y.4480 standard. It is very commonly used in Internet of Things (IoT) applications. LoRa’s low power, long range, and low bit rate make it good for our use to create a long range network of many sensors.

7 Testing

7.1 Verification Testing

Requirement: Wireless Data Transmission

- Status: 100% Complete
- Explanation: Our system uses an Arduino MKRWAN 1310 board with LoRa radio capabilities and a Raspberry Pi 3 board. We verified our system by sending test messages from the Arduino which were successfully received by the Pi board and displayed on the LoRa module for the Pi.

Requirement: Convert RF Signal to DC Voltage

- Status: 100% Complete
- Explanation: Our system requires us to be able to receive a high frequency electromagnetic signal and convert it to a DC voltage value. We tested our system by feeding in a known 2.1 GHz signal from an ADALM PLUTO through a hard-wired connection. We then connected a benchtop multimeter to the output of our PCB and viewed the DC voltage it output. Our system matched the expected values based on component datasheets.

Requirement: Receive a Wireless Signal for Processing

- Status: 100% Complete
- Explanation: We tested this by connecting a low gain antenna to an ADALM PLUTO

and a matching antenna to our PCB. We then supplied a 2.1 GHz signal from the PLUTO and connected a multimeter to our PCB's output. We were able to view the DC value change with respect to the supplied signal strength so we knew it received the signal.

Requirement: Small Size

- Status: 100% Complete
- Explanation: Our system needed to be small enough to be placed almost anywhere. We used custom PCBs which allowed us to minimize the size of our design.

Requirement: System Scalability

- Status: 90% Complete
- Explanation: Our system requires it to be highly scalable in order to create a "network" which covers the area to be monitored. We have been able to connect three Arduinos concurrently and have all three send data to a singular Raspberry Pi and have the data arrive successfully. The system as it stands now can be scaled up with more gateways (also referred to as hubs), and the amount of nodes per gateway could be scaled up with a more elegant network protocol utilizing frequency channels and timers.

Requirement: Cloud Computing Connectivity

- Status: 40% Complete
- Explanation: Our system makes use of Microsoft Azure for handling and processing the data we collect. Currently we have successfully created an Azure hub (also referred to as our "Gateway"), and connected a digitally simulated Pi which is able to successfully send messages containing simulated data. Future developments here would focus on connecting the pi to the internet and reading the incoming message data.

Requirement: System Accuracy

- Status: 0% Complete
- Explanation: We have not yet been able to compare our board's accuracy to a commercially available software defined radio (SDR). We plan to use an ADALM PLUTO SDR and compare the output results from that to the results from our board.

Requirement: Machine Learning for Data Processing

- Status: 0% Complete
- Explanation: Our team has been focused on the hardware side of our project, which includes the data collection PCB and data transfer network. Once our data has been collected and sent to the cloud, we plan on using Azure to host a machine learning model. This model will be used to process the data into a heatmap which varies based on the time.

7.2 Validation Testing

Our project is still in the early stages of development. We are not yet at the point where we would be able to deploy and get external user data from our system.

8 Data Analysis

Our team tested our first prototype board with a hardwired connection to a benchtop signal generator. We had that board tuned to capture signals in the 600 MHz range, which corresponds to the n71 band for 5G telecommunications. Figure 7 shows the ideal output of the commercial RF detector chip in our circuit from the part's datasheet.

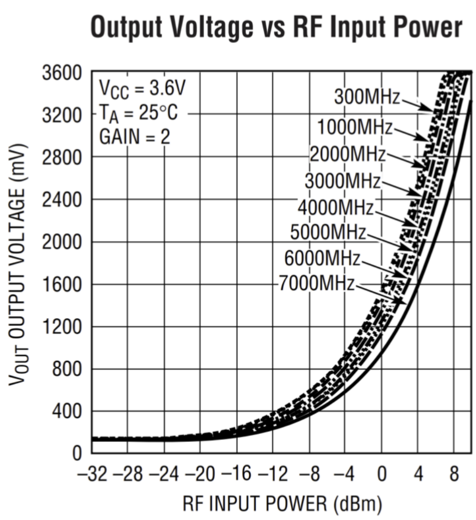

Figure 7. Expected voltage output of commercial RF detector (LTC5530).

We connected the output of our circuit to a benchtop multimeter to view the DC voltage generated from the input signal. We supplied a clean, 611 MHz signal to our board in order to view the output of our circuit with minimal interference from external noise. We started at a signal strength of -32 dBm because our RF detector operates in a range of -32dBm to 8dBm. We then slowly increased the signal strength by a single dBm until we hit the upper end of the operating range. Figure 8 shows the plot our data produced.

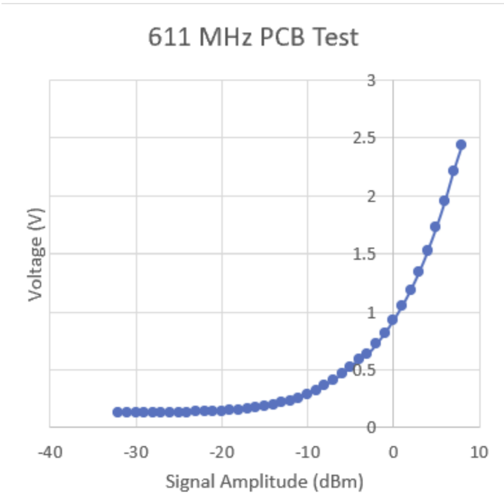

Figure 8. Generated DC voltage plot from PCB.

Our graph differs from the ideal due to a couple of factors. The first is that there could be some internal noise in our circuit which we have not yet been able to quantify through testing. The datasheet graph is also supplied 3.6 volts, while due to the constraints of our Arduino microcontroller we are only able to operate at a maximum of 3.3V. This test confirmed that our circuit was able to produce a DC voltage from a high frequency signal and our circuit operates very close to expected values.

After this test we took a Keysight Field Fox RF Signal Analyzer to multiple locations around the UAlbany ETEC Building to measure the environmental RF signals for the ranges we are interested in. We measured in Mike Denmark's office, next to the window on the second floor, and on the roof observation platform. Figure 9 shows the maximum signal strength values we were able to capture using the Keysight.

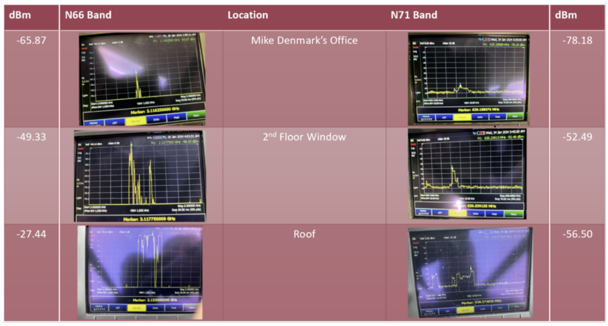

Figure 9. Maximum captured RF signals around the ETEC building.

Using this data we were able to identify the amount of amplification our new PCB required to be able to capture environmental signals. We redesigned our PCB test boards for each band to include amplification. For the n66 band we elected to add a 30dB amplifier to the circuit, and we designed the n71 test board with circuitry to allow for both 30dB and 50dB amplifiers. After testing with these boards, we found that the n66 board with 30 dB of amplification was able to measure signals from cell towers outdoors. At this point we shifted focus to only measure the n66 band, and stopped testing the n71 band. We were able to populate three n66 band RF monitoring PCBs which were used in the final design. The design of this PCB was described in the "Physical System Design" section above

We have also been developing and testing our system's data transfer network. First, we have been developing the half of the system that sends data from the Arduino microcontroller to the Raspberry Pi central hub. First, we connected a single Arduino module to the Pi through LoRa and transmitted messages containing random data. Figure 10 shows the testing setup with one Arduino connected.

Development then shifted to scaling the system to now include two Arduino boards connected to the same Pi to prove our system is scalable. We gave each Arduino a unique node name and timed when messages are sent so that messages are not dropped and lost. The testing setup shown in Figure 11 includes a second Arduino module. Figure 12 shows what the incoming data looks like when receiving

PDF-XChange Editor
Click to BUY NOW!
www.tracker-software.com

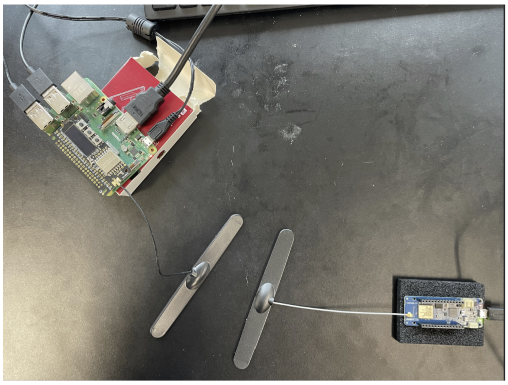

Figure 10. Single Arduino wireless testing setup.

messages from multiple nodes. Currently, we are able to receieve messages from all three of our Arduinos concurrently, and our Raspberry Pi hub is able to plot the data in real-time, while also quality controlling the data by flagging bad values or sensors that have stopped communicating. This data is stored in CSV format on the Raspberry Pi for further processing and analysis.

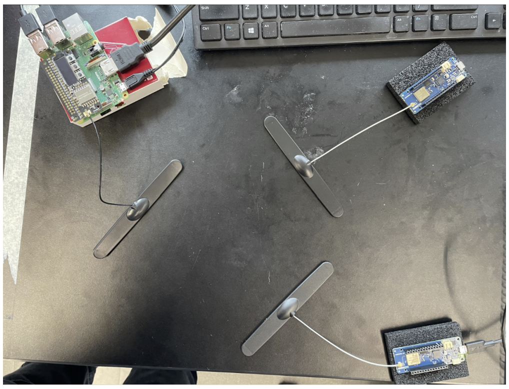

Figure 11. Two Arduinos wireless testing setup.

We have also started developing the second half of the network which sends from the Pi to Microsoft Azure. We used a simulated Raspberry Pi that is available on the web from Microsoft to test this setup to minimize errors due to hardware. Figure 13 shows the simulation we used.

We then set up an IoT Hub on Microsoft Azure to connect the Pi to the cloud. This receives the messages sent and displays the time messages are received, the number of messages, and other network analytics. Figure 14 shows the overview dashboard analytics for

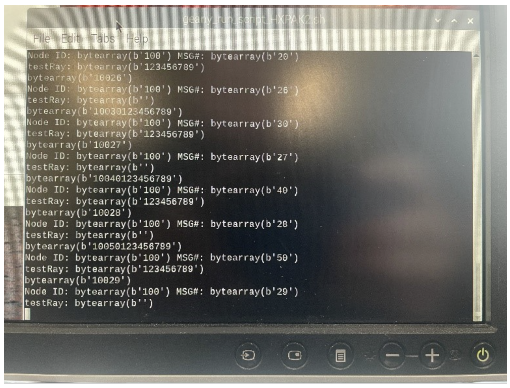

Figure 12. Data stream from multiple Arduino nodes received by raspberry pi.

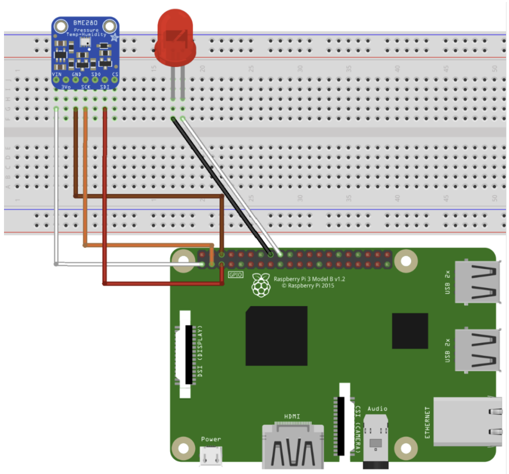

Figure 13. Simulated Raspberry Pi for early Azure testing

the IoT hub.

Currently the endpoints have not been configured in order to be able to view the messages but the dashboard shows the messages are being sent and received successfully. Once the endpoints are configured, we will be able to monitor the actual data being received. We will then shift to converting the system to using a physical Raspberry Pi which we can then merge with the code from the first half of the system to create a single, final data network.

9 Results

Our team began testing by calibrating our final boards and enclosures. We accomplished this by taking an ADALM Pluto commercial software defined radio and establishing a hardwired connection between that and the input

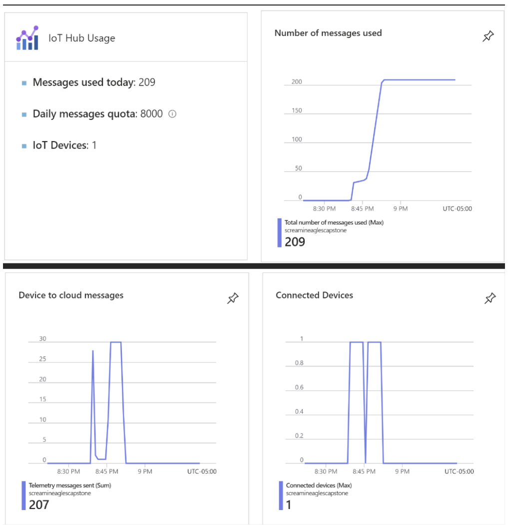

Figure 14. incoming message analytics on Azure IoT

of our enclosure. This can be seen in Figure 15 which shows the Pluto connected to one of the enclosures.

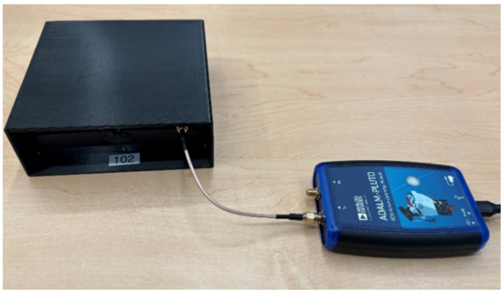

Figure 15. Physical Connection between ADALM Pluto and Enclosure

Once we establish that physical connection, we pass in a noise-free signal of known power levels from -32 to 8 dBm. The values we pass in account for the measured loss in the connections in the enclosure. We are then able to use the measured DC voltage values to create a calibration curve for each enclosure. Figure 16 shows the calibration curves we generated. These curves are stored on the raspberry pi for use in correcting the incoming data as it is measured.

We conducted a two long-term tests of our system in an outdoor environment. The enclosures were placed within 50 feet of each other in an open area with clear sight to two cell towers that transmit the n66 band. Figure 17

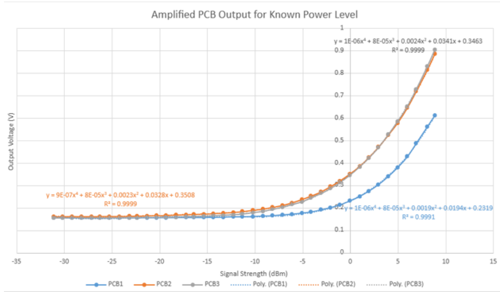

Figure 16. Calibration curves and regression for each enclosure

shows the placement of the boxes. The boxes are covered in a thin plastic bag to help with weather protection, without interfering in the signals being measured.

Figure 17. Outdoor placement of boxes for testing

Figure 18 shows the data captured over a 70 hour period. Two sensors were set up and LoRa connection was established at 11am on a Friday. One of the sensors was disconnected purposely on Sunday to be brought in for a demonstration at the University and was not plugged back in later. This also explains why there is a down period on the other node which was left to collect until 9am on the following Monday, as the raspberry pi was brought out of range and into an area where it was too shielded from LoRa signals. Data reflected the trends we expected to see which showed an increased signal power Friday afternoon and

Monday morning when NYS office workers were in their offices, and much lower average power late at night and during the weekend.

We also conducted a 40 hour test mid-week once we had finished all three of our enclosures. Figure 19 shows the data captured during this run. Again, we saw the same trends appearing as we saw in our 70 hour test, which is that signal power is much lower at night and it spikes during the day, during standard work hours.

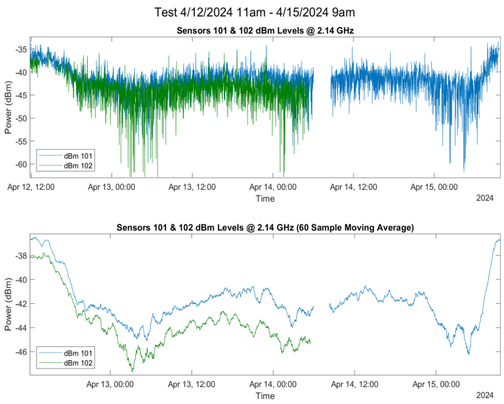

Figure 18. Raw and filtered 70 hour test data

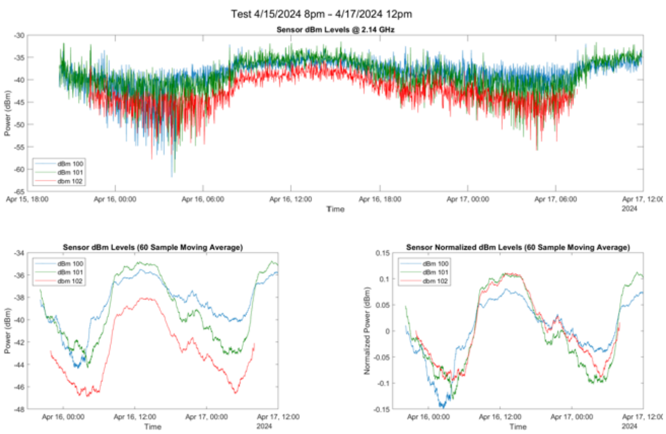

Figure 19. Raw and processed 40 hour test data

We consider these test results positive for a couple reasons. They demonstrate that we are able to leave our enclosures out to collect for long periods of time and they are able to transmit reliably as long as they are given a stable power supply. They also confirm our hypotheses that power would be higher when there are more people nearby using their 5G data for various tasks.

10 Conclusions

10.1 Future Work

Our work has focused on early development and testing of core hardware and networking systems. This opens up several options for future work to continue to build on our project and expand the scope. The first option is to develop a machine learning model using data collected from our sensors and another dataset for some kind of prediction or classification. This is a core requirement of our overall system and design that we failed to meet this year. Another key requirement to develop further is the scaling of the system. Currently our system has three nodes operating off a single gateway, however it should be scaled up significantly larger. This will enable the project to continue real-world testing with reading environmental signals over a greater area and generating data to be fed into the machine learning model. We would highly recommend continued work on the LoRa radio protocol to maximize how many nodes can send to a single gateway, as well as investigating other possible solutions such as LoRaWAN/Iot networks. We suggest scaling down our system even further by taking the core microchip from the Arduino 1310 board and integrating that into the PCB. This would allow our design to be implemented on a single PCB and distributed widely for even greater area coverage. An ethernet dongle could also be purchased a registered to the UAlbany network to simplify adding Raspberry Pi hubs to the internet on campus. We were very happy with our cost-per-sensor, but feel like better RF detector chips could be used with very little cost at scale. We would highly recommend choosing another, perhaps linear response, RF detector chip.

10.2 Conclusions

Based on our early testing our system is feasible and should be carried on. Our network is able to handle three nodes with very minimal data loss. The sensors we designed are able to read signals fairly accurately, and the data we collected with them matches very closely to what we expected the outputs to be. Integration of some of our future ideas, such as the cloud computing and machine learning aspect of the project would greatly increase the use cases of our system, allowing it to be used for long-term research on a variety of topics.

11 Work Responsibilities

- [redacted]'s primary focus was Microsoft Azure work and research. This was not able to be implemented into this year's solution due to issues with registering the raspberry pi on a University network. [redacted] also handled much of the written reports throughout the year and aided in testing.
- [redacted]'s primary focus was to lead the code development for the LoRa network which allowed the sensing enclosures to transmit back to the raspberry pi. [redacted] also handled the data handling on the raspberry pi and creating the live data display.
- [redacted] [redacted]'s primary focus was to design the PCB, including impedance matching between components and creating the footprints for the RF filters in KiCAD. [redacted] also assisted with the coder and network set up and data processing.
- [redacted] [redacted]s primary focus was testing the components used in the RF circuit design to determine that they are working properly and outputting reasonable data. Kihyun interfaced with the ADALM PLUTO and used it to act as a reference for the measurements returned from our system and calibrate our sensors.

References

[1] L. Guénégo, F. Rivet, G. Ferré, A. Walzik, A. Souhayl, and A. Karbab, "A low-cost iot-based device to measure exposure to sub-6ghz 5g waves," in *2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 2021, pp. 1–4.

[2] R. Helbet, P. Bechet, V. Monda, S. Miclaus, and I. Bouleanu, "Low-cost sensor based on sdr platforms for tetra signals monitoring," *Sensors*, vol. 21, no. 9, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/3160