**EIT Vision**

Electrical Impedance Tomography (EIT) that determines an object’s location using Radio Frequency (RF)

Sponsors: Andy Goldstein and Eric Bogatin. TA: Hardik Minocha

Team: Wavegen Warriors

Project: Imaging with EIT System

Dante Ausonio, Peter Braza, Ahmed Khan, Gustavo Sanchez, Graham Snyder, Gavin Unrue

### Table of contents:

Team Roles 2

Product Overview[.](https://docs.google.com/document/d/176O-6nSo4TlcZtWL-M44HITeT-06Gjq8sbbAPpwVq34/edit#heading=h.rr6xg8pj86sv) 3

Engineering and Marketing Requirements4

Technology Options5

System-Level Design6

Product Analysis7

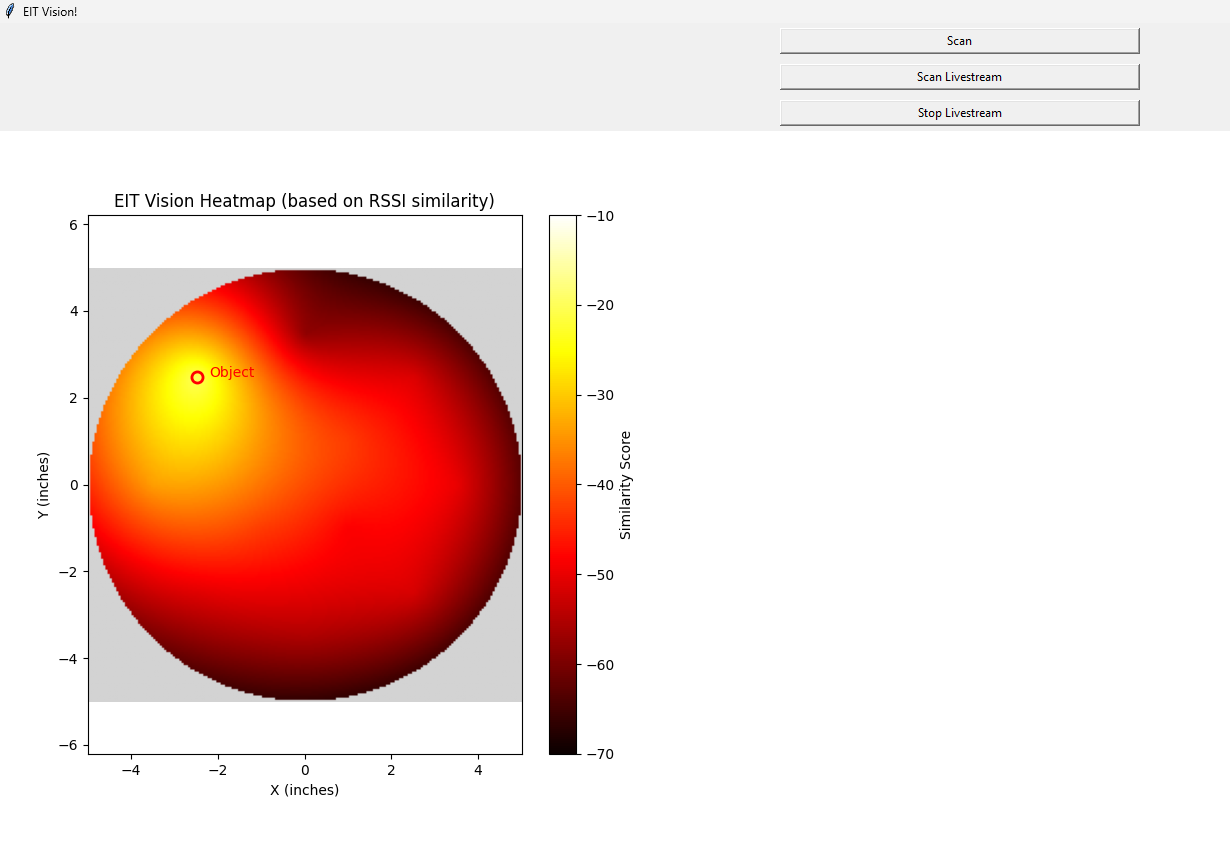
Learning through the Project8

Individual Reflections11

**Team Roles**

| Team Member | Lead Management Role | Backup Management Role | Lead Engineering Role | Backup Engineering Role |
| --- | --- | --- | --- | --- |
| [Dante Ausonio](mailto:anau1488@colorado.edu) | Team Lead, Scrum Master | Logistics Manager | System Architect, UI Development | Software Engineering |
| [Peter Braza](mailto:pebr3960@colorado.edu) | Customer Manager | Project Schedule Manager | Embedded Systems | Power Management, Testing, Analog-Debugging |
| [Ahmed Khan](mailto:ahkh8808@colorado.edu) | Communications Manager | Morale Manager | PCB Design, Analog-Debugging | RF Design, System Architect |
| [Gustavo Sanchez](mailto:gusa9787@colorado.edu) | Project and Assignments Schedule Manager | Communications Manager and Team Lead | Test Engineering, Power Management | Power Management, Analog-Debugging, Software Engineering |
| [Graham Snyder](mailto:grsn1584@colroado.edu) | Budget/Logistics Manager | Customer Manager and Team Lead | Hardware Enclosure, RF Design | Analog-Debugging, UI Development, Power Management |
| [Gavin Unrue](mailto:gaun8195@colorado.edu) | Morale Manager | Assignments Schedule Manager | Software Engineering | Embedded, PCB Design |

**Product Overview**



*Figure 1: EIT Vision Product (Left), Imaging Output (Right)*

**Elevator Pitch:**

The EIT Vision project is a research project that verifies the feasibility of using radio frequency (RF) signals for Electrical Impedance Tomography (EIT) imaging, offering an alternative to the traditional methodology of using direct current (DC). EIT Vision generates a 2D heatmap that visualizes the location of a conductive object placed within the 10 inch circular platform. To accomplish this, it transmits and receives RF signals between six transceivers with antennas arranged evenly along the circumference of the platform. The system measures the changing power strengths due to signal reflections or blockage from the conductive object, enabling image reconstruction and proving the proof-of-concept.

**Future Possible Implementation:**

RF Imaging is a growing field that has been explored, but the EIT Vision implementation focusing on accessible, off the shelf LoRa modules, and combining them into a circular plate allows for future imaging options. The EIT Vision works effectively to locate conductive materials, like a metal detector. It can be used in a larger system to detect landmines underground, an alternative metal detector, or future healthcare imaging applications. Because EIT Vision has non-ionizing radiation, future imaging applications of the device could provide a safer alternative to current invasive imaging methods.

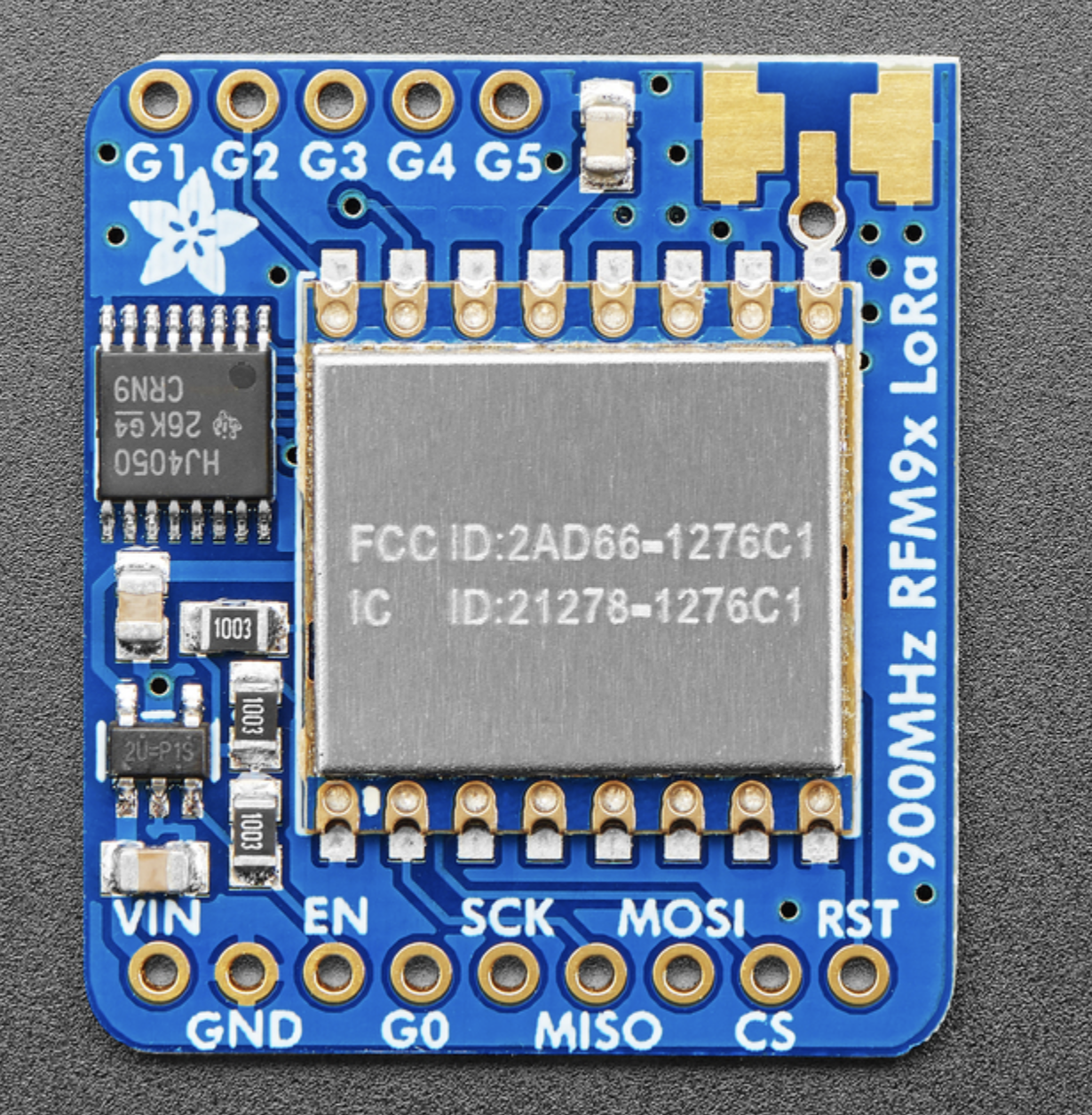
**Marketing and Engineering Requirements**

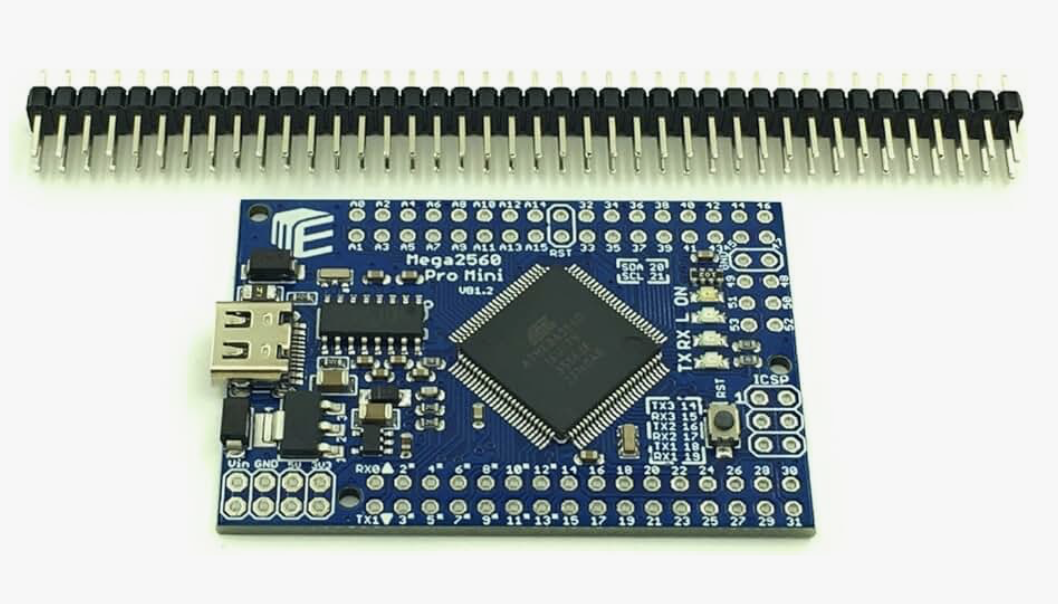
The marketing requirements for EIT Vision emphasize usability, speed, and accuracy from the customer’s perspective. Users must be able to place various conductive objects onto the sensor platform, and a clear and accurate 2D image of the object must also be displayed, with all results delivered within 30 seconds of initiating a scan. To enhance accessibility, the device should offer simple, intuitive instructions and output, allowing users with no technical background to understand and operate it successfully. Additionally, automatic sensor calibration on startup ensures accuracy, and while not essential for the minimum viable product, features like support for multiple objects, low power consumption, and a one-foot sensor ring diameter are noted as desirable enhancements.

To meet these marketing goals, corresponding engineering requirements have been defined. The core functionality is driven by an EIT (Electrical Impedance Tomography) system that uses LoRa modules to detect AC signal variations caused by objects on the sensor ring, which are then processed to determine object classification and shape. A responsive user interface facilitates user interaction and displays results. The system must be capable of delivering at least one processed frame within 30 seconds, and sensors must operate sequentially to ensure accuracy. The device must perform a power-on self-test to calibrate sensors and mitigate environmental interference. It is powered by a 5V line and is designed to operate efficiently within a ±15V, 2A range. A minimum of six sensor detector nodes is required to produce a reliable 2D image, though more sensors may improve resolution. Together, these specifications ensure that the device aligns with user expectations while remaining technically feasible.

**Technology Options**

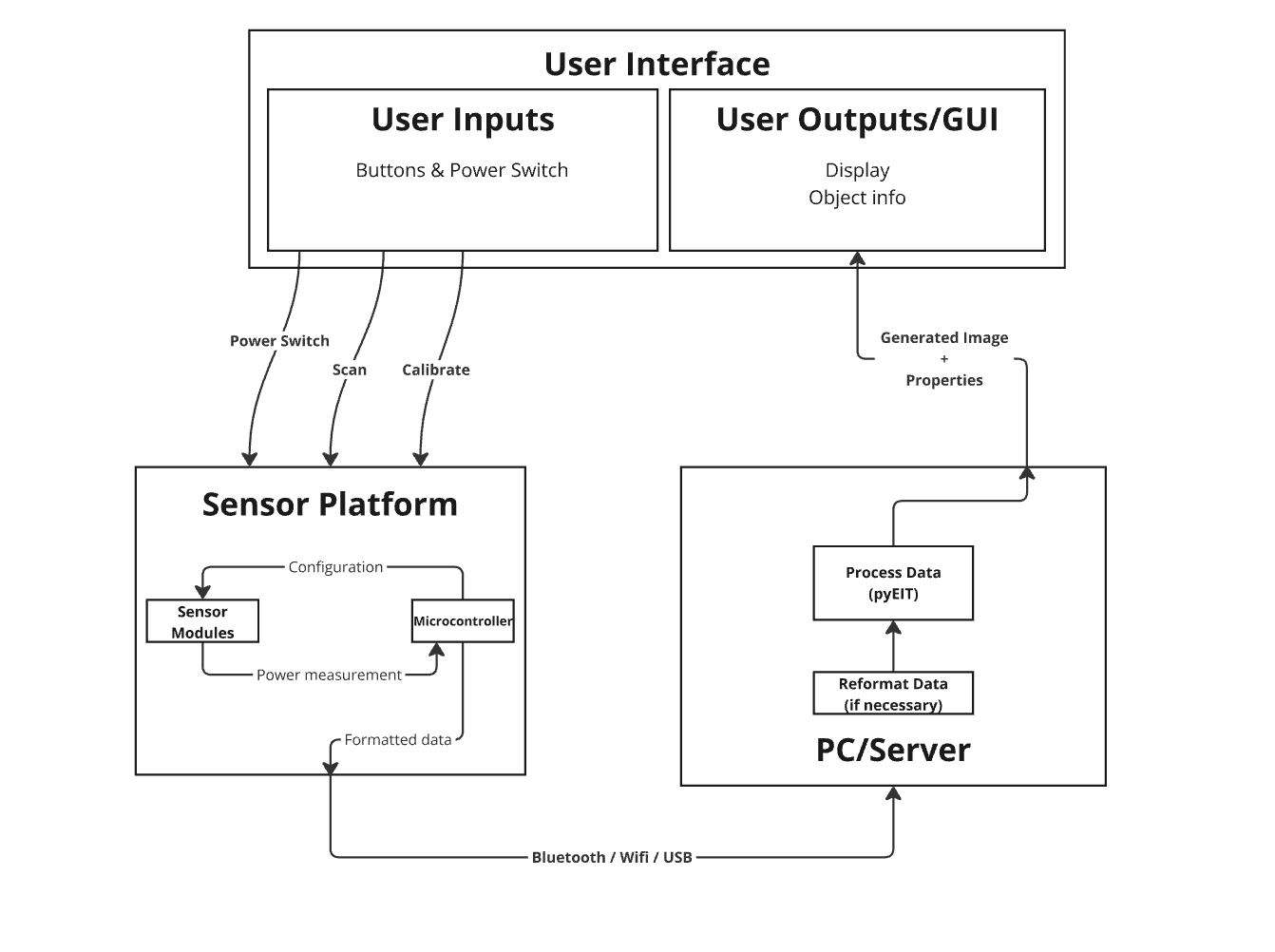
* RFM95W LoRa transceiver modules



* These are the main LoRa transceivers which we used on the platform and operate at 915MHz. We used these by transmitting a single character ‘H’ and then measured the RSSI using the other five modules.
* Arduino Mega 2560 compact version
* This microcontroller was chosen because we initially were concerned about the Arduino UNOs not having enough interrupt pins to properly configure and operate even 4 LoRa modules off of the same microcontroller. The Mega stood out to us since it included six integrated interrupt pins without needing any additional configuration. Of course it is technically possible to configure some of the digital pins into interrupt pins, but due to time constraints we felt that doing so would add too much unnecessary risk. We initially were using the standard size of arduino, but we found it was a bit cumbersome to implement as part of our system, and since some of us already had a bit of experience using this more compact version of the mega, it seemed like an obvious choice.
* Arduino IDE
  + For use with programming the Arduinos.
* Python
  + Radial Basis Function Library: Performs the actual similarity computation to generate the heat map of the object placed on the platform.
  + Tkinter UI: User Interface framework for setting up the buttons, and display. Sends input signals to PySerial, and reads outputs from PySerial. Programmed to automatically perform scans every 2 seconds, and update the displayed image accordingly.
  + PySerial: Communication between the Arduino and Python Image Processing Script. Handles signals for performing scans, and reading scanned data.
* Custom PCB Wiring Harness
  + The two main purposes of the PCB were to organize all of the many connections between the microcontroller and the LoRa modules and to create a solid and repeatable foundation for our testing since the modules are directly plugged into the board at the correct locations for the entire platform assembly simplifying construction. Having a reliable and repeatable test setup also helped immensely as we had to often move the platform and perform many tests in different conditions and needed a more reliable way of doing that where wires wouldn’t just fall out like our earlier breadboard setup.

**The How**

**Functional Block Diagram:**



**Describe how you implemented some of the key functions and how they interacted with each other:**

We started our plan with having a central microcontroller that would manage all of the LoRa configurations as they need to be repeatedly reconfigured for the six different patterns and it would also be used for data collection and transmission. Our initial impression was that the arduino and most other microcontrollers would not be able to perform the EIT computation since we were initially planning on using PyEIT and EIDORS which are more computationally heavy libraries. With this assumption in mind we decided to use an external PC that would create the images with the formatted data from the microcontroller.

The user starts by either pressing the button to do a single shot scan of the platform, or the live view mode which constantly updates the image approximately every two seconds. This is recorded by the UI which then sends the corresponding mode command to the Arduino which then begins the scan. The Arduino goes through the sequence of six configurations by communicating with each of the LoRa modules over the SPI bus integrated on the main PCB, where they are all set to either send or receive a single character and then measure the corresponding RSSI value. The data is then sent as an array containing all 30 of the data points from all six configurations and is preprocessed with a simple mode selection algorithm to clean up some of the noise to the external computer where it is then processed using the Radial Basis Function to create a heatmap of the space between the antennas. The algorithm also marks the spot on the heatmap with the greatest intensity as the possible location for the cylinder. All of the communication between the Arduino and the external PC is sent over usbSerial at a 9600 baud rate. The entire platform is powered directly from the usb connection to the external PC which is powered from the wall making the whole system fairly portable.

**Analysis**

We met a lot of the requirements, but missed a few. For what worked, we completely surpassed the one image every 30 seconds requirement. We were actually able to get it down to just 2 seconds per frame with live feed updates. Another success was the group’s ability to utilize a similarity based heat map library, Radial Basis Function, to show dynamic location images accurately representing the location of the object on the platform regardless of where the object is placed. We were even able to see two distinct shapes when we placed two objects on the platform.

What did not work was using pyEIT or EIDORS for the imaging. These libraries were too specific to the DC implementation of EIT to be applied to our RF implementation. If we had more time we would likely design our own algorithm to perform imaging in an analogous way to pyEIT and EIDORS. Another shortcoming of our system was that it was unable to classify objects based on their dielectric properties. This was always a stretch goal anyway, but it was one we did not achieve.

**Learning through the Project**

* We learned about the importance of reading documentation before going too far down the development path.
* We learned that EIT Imaging can be performed using RF.
* We learned to communicate without interrupting each other.
* We learned how to integrate a GUI for a project.
* We learned how to program and use LoRa.
* We learned about various imaging techniques and libraries.
* We learned how to professionally communicate with people involved with this project.
* We learned how to manage our time effectively.
* We learned how to write technical documentation.

**Team Reflections**

**What did you learn from this experience?**

**What were some of the highs and lows?**

**If you had to do this project again, what would you do differently?**

**How will you approach complex, team projects in the future based on what you learned in this capstone experience?**

**Dante Ausonio**

I learned a lot from this project. From the fundamentals of RF, and how electric impedance tomography works**,** to how to work on a team, this capstone project greatly enhanced my engineering skills, and helped me become a well rounded engineer. One of the most important skills I learned was managing communication on a team. In our conversations I often found us interrupting each other and small side conversations branching off from the main conversation leading to disconnects between team members. To address this, I had a conversation with everyone addressing our tendency to interrupt one another and branch into our own conversations and the problems this leads to. I expressed that it’s natural to do so, and it's not bad to interrupt people, but to just try and notice when you are interrupting and back off. I also told everyone that it is entirely justified to call someone out for interrupting you. It’s not a criticism, rude, or aggressive to do so, it's a natural human mistake that happens and just needs to be called out so everyone gets the time and space to communicate. I was no exception to this, even though I noticed it, I often caught myself talking over people and needing to back off so they had the room to speak. This improved our communication because everyone was able to speak their mind without feeling rushed, and we each became better listeners by holding ourselves back from talking over one another. To address the branching conversations I simply tried to call people back to the main conversation whenever I saw one happening, and then gave them an opportunity to speak after the current speaker was done, because these branching conversations would often happen when someone had an idea, but didn’t have a platform to express it to everyone, so they would just share it with one person on the side.

There were many lows throughout the development of EIT VIsion. At the beginning of the project we were struggling quite a bit because we were working on a research oriented project, and the assignments were geared towards engineering projects with more known factors. For example, we were expected early on to have a tech stack that we would know would work, but the whole point of our project was to see if this tech stack would work. So when assignments asked things like ‘how do you know this will work’ we would respond with ‘we don’t know because that is the point of this project’. We often went on unnecessary side tracks to try and prove something would work when ultimately there was no way to know until we actually put the full platform together. For example, we were often asked to try and characterize the noise we might expect to see once we put the platform together. Or we were pushed to design a faraday cage to minimize noise that was ultimately useless and wasted a week of production time because once we just actually built the platform we were able to show that noise was negligible. All of these side tangents, and mis-aligned assignments lead to a lot of frustration on the team due to our slow progress.

On the other hand, there were many highs as well. The highest high for me followed after one of our lowest lows. About 2 months before expo we ran into a software limitation on the number of interrupts we could use on our microcontroller. The hardware had 6, but the software would only allow us to use 3. Unfortunately, we needed at least 4 interrupts for our system to work. This was a huge limitation that put us in a situation where we would need to re-design our entire system. In our panic we developed multiple alternatives, but each was going to demand quite the development process. Fortunately Peter investigated the package we were using, and was able to navigate through the file system to the exact lines of code that were creating that limitation and change them. Peter and I came in late one night to test this solution and after a little debugging we were able to get it to work. This was quite the emotional rollercoaster, and the euphoria we experienced the moment we saw all of the interrupts working was incredible, and celebrating this victory with the team felt great.

If I were to do this project again I would have followed a more direct path to the end goal, and resisted the urge to go down all of these unnecessary rabbit holes that were ultimately unnecessary. I would also push the team to use GitHub instead of just a long list of drop down menus with different versions of our code. For future team projects I will be sure to do more team binding sooner. I think once we started laughing together and connecting we started to work together a lot better.

**Gavin Unrue:**

**What did you learn from this experience?**

I learned quite a bit about effective team communication and about working within multiple teams all doing different tasks in the same project. That made me a little more conscious about

writing my code with clarity in mind so that others have an easier time understanding it.

Otherwise everything was pretty straightforward, I've had experience with projects in the past and this wasn't much different.

**What were some of the highs and lows?**

One of the lows was learning that the team decided to use PYEIT after I had spent

a few weeks investigating EIDORS. Another low was when our team got the first sensor

prototype working and we learned that our progress with PYEIT was all for nothing since the

data did not agree well with the algorithm.

A high I would say is when I got the data pipeline working and was able to read data from the hardware and port it into the software. It was good to know that the code I wrote worked well.  
 Another high was when graham found a different functional algorithm which was able to

produce a heatmap from the lora sensor data readings.

**If you had to do this project again, what would you do differently?**

If I had to do this project again I would invest more time in the

research phase of our project. Since the algorithm that we selected was not well suited

towards our application, we could have saved lots of time and energy that was spent

troubleshooting EIDORS and pyeit towards improving our product.

**How will you approach complex, team projects in the future based on what you learned in this capstone experience?**

In the future I will write code to be as clear and coherent as possible so that others have an easier time understanding it.

**Gustavo Sanchez Sanchez Jr.:**

**What did you learn from this experience?**

After having worked with the same six people for two semesters, I learned what I need to improve on, interpersonal skills, and technical skills.

From myself, I learned that I still have that annoying mindset of only focusing on one task without paying attention to my other tasks. I realized that this is something I have to be conscious about when I start my career since I will be in charge of completing multiple projects simultaneously. I also need to improve by letting go of my ego in times of need since I refuse to ask for help and hope to complete objectives on my own; another characteristic I need to be conscious of in my career.

With interpersonal skills, I learned how to effectively communicate with the team and remind teammates of important deadlines. Aside from having to be responsible for my other school work and personal life, having had the role as the “reminder” person made me feel as an important factor to the flow and completion of important deadlines. From what I remember, I was on top of my role by sending reminders three days before the deadline and not once did the team miss a deadline. On top of that, I also learned how to communicate with “higher” positions within the group and give pointers/feedback to teammates in a way where it did not seem aggressive or passive aggressive.

On the technical side, I learned how to integrate and debug code for a GUI. This was my first time using a GUI and it was beneficial to see which libraries work best to develop GUIs, how to debug them, and how to design the flow of the external code with the GUI. In addition, I learned how imaging software works and how to build and debug respective imaging libraries.

**What were some of the highs and lows? If you had to do this project again, what would you do differently?**

One of the highs for this project was having integrated the PCB with the four LoRa modules successfully. Before this had occurred, for months prior our prototype consisted of only two LoRa modules with messy wires everywhere. We were consistently told that we were behind compared to the other teams so when we succeeded in getting the PCB to function with the LoRa modules, we felt accomplished and confident. Another high was when Graham got our imaging software working. After having tested the pyEIT imaging software and determined it would not work with our project, Graham saved the day by finding an alternative library and integrating it with our system within hours.

One of the biggest lows that bummed me out for a while was the fact that Gavin and I had spent months researching and implementing our code using the pyEIT library, just to evaluate that the code would not work with our implementation. It sucked to not have within our final product but we learned that for next time we need to think smarter and not harder.

If I were to do this project again, what I would do differently would be to effectively evaluate the imaging libraries before committing to one. I realized that in this aspect, I could have found one library and done everything in a timely manner and then spent the rest of the time helping the team with the hardware.

**How will you approach complex, team projects in the future based on what you learned in this capstone experience?**

For the future when I am involved with complex, team projects, the three main points I will use to approach these projects will be to establish roles and responsibilities within the team, ask questions when I need help, and effectively spend time on my portion of the project.

**Ahmed Khan:**

**What did you learn from this experience?**

Throughout this experience, there have been many things that I’ve learned, especially the greater variety of skills that I have now both among the more managerial side of things like the documentation, the design review presentations, and the MRD/ERD, but also skills that are critical to working in a team. Skills like managing expectations with our timeline and making sure that we avoid scope creep while also satisfying necessary requirements.

When it comes to more specific skills, I learned a little more about what goes into designing large PCBs and how that impacts both manufacturing costs and the final assembly of the project. This project also helped reinforce the lesson that the hardest thing about a project or parts of a project is finishing, as I was getting sidetracked early on which did slow down our progress when it came to transceiver selection and by extension the PCB design schedule. I also learned that the best way of avoiding the bad decisions made by others is to build it yourself since that was the source of many problems when it came to the LoRa modules and the libraries that run them.

**What were some of the highs and lows?**

One of the great moments this semester was when Graham improved the algorithm with the RBF library and made the platform generate actual images instead of our backup plan which we had already resigned to using since that plan felt like too much of a compromise on the functionality of our project. Another great moment was when we found out why our attempts to use more than two modules kept failing, since we had been stuck on two modules for well over a month and kept getting reminded that we were behind. It finally felt like our first real jump in progress for the project and gave us something tangible to show. Of course Expo was great since everything went as smoothly as we could have hoped. There were no significant problems with the platform for the entire day and it was really easy to do demos with it since the sensors gave instant feedback.

Some of the lows included when we ran into significant scaling errors with the LoRa modules. Every time we would set up our two sensor test, it would cause the value to jump between -20 and -45 dBm and we couldn’t come up with a solution other than shifting our values to avoid that region of the measurement. We tried many alternative solutions, and even considered switching to different modules, but due to time constraints we were unable to explore them in any significant depth and that also slowed us down since we were not sure if this error would impact the final platform. Eventually we found an alternative to manually changing the data by switching to the properly sized antennas for the modules which by increasing the output power and input sensitivity, raised our RSSI values above the nonlinear region completely avoiding the problem for our specific use case.

**If you had to do this project again, what would you do differently?**

If I had to do this project again, I would choose a different module, preferably a variable frequency generator, and design the receiver circuitry myself. I would also spend far more time investigating the setup from a conceptual and physics based view since some of the problems we encountered were due to a lack of understanding about what was actually going on with the RF signals in detail in the platform area. This would also be improved with the use of actual simulations since that would allow for better fine tuning of the imaging algorithm and debugging.

**How will you approach complex, team projects in the future based on what you learned in this capstone experience?**

Looking forward, I will be making sure that whenever I work on a team project all of our major goals are well defined and planned in advance so that we reduce the risks of getting sidetracked.

**Graham Snyder:**

**What did you learn from this experience?**

Throughout this project, I overall learned that a long term project over the course of an entire year is almost never going to go as planned, and there will be moments where the finish line seems near impossible, yet it still can be done by overcoming challenges. From the start of this project in October, to the end in April, the everyday work of the project drastically changed. It started from nobody in our group having a great understanding of RF, to then being able to create progress at such fast speeds in March and April that we were able to get the final vision of full imaging working for Expo.

From the start of the project, we did not know where to start. We tried doing lots of research about traditional EIT, and the product guidelines were a bit vague. Looking at when we made the most progress as a group, it was the times where we put down the research, lecture notes, or even knowledge shared by professors and TAs, and just started to explore ourselves. I really was able to grasp a significantly better understanding of RF, more than the wireless systems class I took sophomore year. Being able to come up with our own experimental tests based on limited knowledge, find new outcomes, and then confirm through online research our findings was really cool to see in terms of learning. I learned that sometimes you need to find the knowledge yourself and through your current understanding rather than researching online. Online lecturers and Youtube videos have no idea what your baseline knowledge is.

In terms of group work, I overall learned the significance of having set roles, two week sprints, and daily scrum. Starting a project with Expo seven months away is daunting. I remember always thinking before taking capstone how a group of six members can structure that much time. Of course, the class has its own structure for tests, presentations, and mini-expos, but I learned how creating your own timeline is really effective to get a ladder connected to that final Expo date. Even if the timeline is completely wrong in the end, having a very detailed vision to success is so helpful when you are in a big group.

In terms of content, I know so much more about RF, microcontrollers, PCB design, 3D printing and laser cutting, Python and Arduino code, LoRa sensors, and other parts of this class that would take multiple classes combined to learn.

When working with sponsors, I learned how important a well-defined MDR and EDR is to effectively communicate the same vision. The vision is always adapting in everyone’s mind, but laying it out clearly and written makes it so much easier for that pathway to Expo.

**What were some of the highs and lows?**

I would definitely say the lows include the frantic mess of finding the transceiver to use in the final product. It took way longer than it should have to get the first LoRa sensors working, and they were not great. Again, with our lack of RF knowledge, we didn’t realize how to initially fix issues such as low sensitivity and the gain issue. We then lost track of our vision. We went and bought many LoRa sensors, and even tried building our own custom oscillator to find better data. What we should have done is learned why our initial LoRa sensors were not working as intended. Was it noise? Is it built into the header files? Can we hard code the power? We just gave up, and moved onto other modules, and that cost us very valuable time. Once we caught on and decided as a group to just pick one module and go, no looking back, that is where we made leaps in progress.

A high point of the project was when Peter figured out how to get multiple LoRa transceivers working on one Arduino by adapting the built-in RadioHead libraries. We started to feel disappointed and hopeless mid-February when only three sensors could work. At this point, our resolution could never be good in time for Expo. However, instead of moving on to other modules or something that would restart the timeline, we kept persisting for two weeks, and after so many oscilloscope tests, code debugging, etc, Peter finally found the issue and just like that we had a clear light at the end of the tunnel for Expo. That’s when it became exciting, as now we could come to class and work on a functioning basic prototype each class. Looking at that data each class is where I learned so much about RF. Ahmed also creating the best antennas helped so much with the initial sensitivity issue.

**If you had to do this project again, what would you do differently?**

I would’ve gone full hands on with modules from the start, and done way more focus on analyzing the data from the LoRa sensors to learn how to fix issues. I still was not as confident with RF signals and how conductors distort the field with reflections, and I really learned that when I was able to see the RSSI values in real time show that off. If we did minimal research, got hands-on with multiple LoRas from the start, and performed as many tests with as many objects as possible and scenarios, then we could’ve had a much faster timeline. I think overall, there was a disconnect between hardware and software for a while. The RSSI data is never going to match EIT imaging with DC currents, and I think we should have spent more time teaching both sides of our group what we were seeing. We probably would have found a different imaging algorithm sooner.

**How will you approach complex, team projects in the future based on what you learned in this capstone experience?**

Team projects are complicated in that everyone has a different level of knowledge, so everyone needs to be caught up to date each day. It may seem like a waste of time, but it is so important that firstly you teach your knowledge, reinforcing it, and secondly it prevents wasted time where there was a disconnect in project work. Creating a full timeline to the end deadline is also super important. It may be completely wrong, but if it is a live document, you can keep updating it, and use it as a way to show progress, whether that is failures or successes. Also, separating the group into roles, but not being too strict about it, will really help as well.

**Peter Braza**

**What did you learn from this experience?**

**What were some of the highs and lows?**

**If you had to do this project again, what would you do differently?**

**How will you approach complex, team projects in the future based on what you learned in this capstone experience?**

**What did you learn from this experience?**

There are so many things I learned from this experience. The first is realizing that connecting with your team is very valuable in order to figure out how each person functions optimally and so that you can properly motivate each person to work (if needed). It’s not necessary to be friends with people on your team but it is important to have some level of camaraderie established. Too much friendship and work can be unproductive, but a healthy middle ground between coldness and effusive friendship is best. I also learned to appreciate the schedule set in place by the capstone staff; for example, the TA check ins once every two weeks. This helped keep myself and others accountable and it seems like something that could reasonably be encountered in industry.

One skill I already knew at some level but came to appreciate even more during the course of capstone was the importance of the trait of perseverance. This was important time and time again, from getting the Loras working for the first time to debugging a certain piece of code. For me personally, though, this was extremely important in getting all six, or even more than four Loras functioning at the same time. There was a point in time where every time that we tried getting more than 3 Loras working on the same board, we would simply get a “wiring error” even if the wiring was perfect. It had to be a software issue. What I ended up doing was persevering through reading painful Header and CPP files in order to hack them to make our board work with four and then six Loras at once. Ultimately, this worked, and this was a significant reason we were able to have more than three Loras functioning at once. This ended up being quite important to our project’s functionality. While I can’t honestly say I learned anything new hacking these files and adding, say, new interrupt service routines, but this was a good refresher of the embedded course I took, ECEN 2370. This indirectly taught me how important certain classes are and more than that; being a bit of a jack of all trades in EE helps very much with success in project settings.

**What were some of the highs and lows?**

This year was all over the place. One of the biggest highs was when Dante and I tested and finally got four Loras working on one Mega. This changed the course of our project for the better; it was extremely exciting. To oppose this, a low was debugging four Loras in the hardware and the Arduino side of the software for quite a long time to no avail. This made the high of seeing four Loras function for the first time even better. Another low was our PyEIT imaging algorithm not functioning whatsoever. We had feared this might be the case since PyEIT is designed to work on DC systems and of course ours was an AC system; it was fundamentally different. We also knew that PyEIT did not fully utilize the data we gave it. Even so, though, it was painful watching it not function whatsoever. To counteract this, a high point of the project was when I saw the RBF library working for the first time to generate a heat map of where our object is in the enclosure. Graham did great work to get that functioning, and Dante too in getting the UI to update in real time. Another low was working hard on implementing a memory efficient and computationally efficient algorithm to calculate the variance and covariance of our signals but then not using this at all in the final design. While I have some hope that a future group could use this code to get a better statistical understanding of our system, it is likely I will not ever see that work being done (if it happens at all). Yet another low was not being able to solve (without some hacky nonlinear filtering) the issue of the power reading jump. A high that somewhat opposed that low, though, is when Ahmed changed out the antennas to ones that were better matched to our wavelength.

**If you had to do this project again, what would you do differently?**

There are undoubtedly things I would do differently if I had to do this project again but that could take many different paths. First of all, I would have the team work on developing a custom oscillator or at least use transceivers that run at a lower frequency. This is so that we could stay in the near field and so that we could more easily recover the phase of the signal without extremely expensive dedicated hardware. The phase information would tell us much more about the location of the object but this was, in the context of our limited project scope, virtually impossible to collect at 915 MHZ. This has another benefit of that with the Loras that we used, we only had a resolution of an integer of DBM. Getting at least hundredths of a decimal precision of DBM would be quite helpful. Another major change I would implement is to rotate the object or the sensor platform around the object. While I liked the fact that we could generate an image without this rotation, rotating the object would give us much more information per sensor. We also would need less sensors compared to not rotating the object. Yet another change I would implement is to add a Maximum likelihood filter based on the covariances of the noise of the transceivers in our enclosure. Yet another change would be to simulate this system using HFSS software and get a good idea of what we expect to see. Also, it would be cool to try to do object identification, whether this be done analytically with physics or through Machine Learning algorithms. Also, depending on the frequency that we use, I think it would be wise to add radar absorbent foam to reduce sources of outside noise.

**How will you approach complex, team projects in the future based on what you learned in this capstone experience?**

In the future, I honestly don’t think I will try to approach team projects much differently than what we had going this semester. Ultimately, we all worked together well without any major issues. At the beginning of my next complex team project, I may (depending on the setting) ask the others to honestly list their strengths and weaknesses so that we can separate work optimally. I’ll also try to connect with the others in my team while not forcing friendship. I will also strive to persevere through any difficulty and encourage others to do the same. Having a timeline would be very helpful as well.