TCAT Team Project: Implementing a LoRaWAN-based Bus Tracking System

Michelle Davies, Andrew Lin, and Kira Weinberg

Cornell University, *Internet of Things* (*ECE4950/MAE4220/MAE4221*)

Professors Max Zhang, Stephen Wicker

Spring 2021

Executive Summary

Our vision for our project was to create a LoRaWan solution for TCAT to send their data from busses, to their central server. The challenge that we were addressing was to see if LoRaWan is a valid option for TCAT to move towards, and away from cellular which would save them money as well as help build the early open source infrastructure for other companies to access affordable internet connectivity for their projects. For our project, We designed a prototype that could take its own GPS data, and send it through LoRaWAN and then map the gateway coverage of TCAT routes. This project proves the early potential for developing a LoRanWan system for communication for TCAT's busses and other companies that might deploy very similar solutions. Once we collect the data, it is plotted on a map to see where the data was successfully sent, where the busses drove, and how fast the bus was moving at the time so that we can see where coverage exists. In our project we have mounted one prototype to a bus that is able to drive any routes requested. This will help us to determine the effectiveness of our prototype for our application.

Our physical prototype consists of a 3d printed box that contains the devices that are mounted inside the bus's electronics cabinets, is powered by the buses themselves, and is wired up to the top of the bus with an antenna shield/dome to protect the antenna from being damaged by the daily bus wash. While the device is in field testing, we will find out if the shield can withstand the bus wash, and also if the antenna is able to transmit data through the shield or if it needs to be redesigned to be less limiting. The final vision of the mechanical box is one that is simpler to use, heat resistant, and able to protect the electronics inside and the antennae while minimising their impact on their functionality.

The next steps in this project's research and development would be to integrate the real information management system, Popufare, onto the device as well as connect it to the TCAT's central server. From there, the web interface that displays the data could be hosted on a server in TCAT's network so that the results can be updated in real time and viewable to everyone with access to the internally-secured TCAT work network. The data could also be transmitted to the TCAT central server so that external applications such as Google Maps can access bus positions. We were only able to conduct field testing for half a day, but the system is wired up on the bus and is able to continue running, so this information will be available to whomever takes over the project in order to finish mapping out the coverage of local LoRaWAN gateways. From there, we would be able to see where there needs to be more coverage, and potentially work with the city to see about adding supporting gateways to those areas, or change the prototype to have better connectivity. We will also learn about the various environmental factors that may affect our prototype's function from geographical challenges such as cliffs, to weather.

Keywords: Lorawan, TCAT, Public Transportation, Internet of things

Implementing a LoRaWAN-based Central Communication System

Background and Social Context

Bus transportation is a fundamental public service and a socially conscious and economical way to travel. The Tompkins Consolidated Area Transportation bus service, known as TCAT, serves the Tompkins County population of roughly 100k people including both students of the local universities and residents, and does so in a reliable and efficient way. In the year 2019 TCAT operated 63 busses forming 32 routes that drove a combined total of 1.735 million miles and gave 4181555 rides. TCAT serves a large student population, as 53.91% of the rides taken are by students [19,20]. It is a very large and robust transportation system that is integral to the communities that use it.

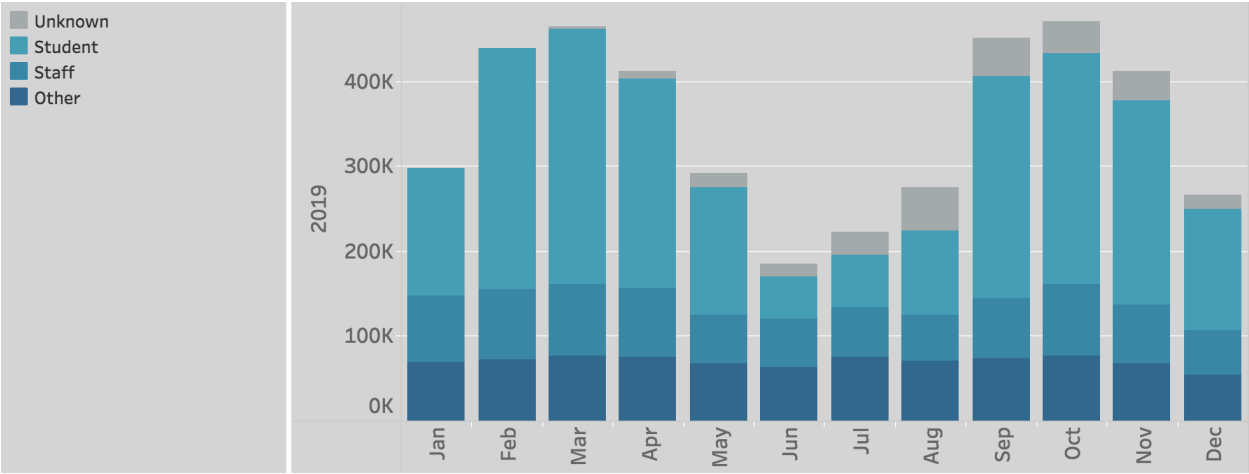


Figure 1. Bus rides in 2019 by type of Rider. [19]

The TCAT uses a fare collection system and GPS tracking that would greatly benefit from being shifted to a Lorawan solution. In order to function properly, the busses are tracked via GPS so that passengers can reliably know where the busses currently are, and the busses also must send fare collection data to the central server. In our project, we are working closely with TCAT to design a first prototype that would connect TCAT's fare collection system, Farebox, to TCAT's central server through a low-power wide-area network (LPWAN) as well as GPS tracking information. This would increase coverage in a sustainable, affordable, open source, way and build the technological foundation for future innovation using the system.

Currently the fare collection system that is used by TCAT is on cellular. It is not ideal as it is very expensive to maintain, costing around $30,000 annually to uphold [3]. This is a problem because the high cost and complexity of the set up bars smaller companies with similar needs from being able to set up their own systems. We will be using LoRaWan, a proprietary modulated LPWAN with a Raspberry Pi. What makes LoRaWan cheaper than cellular is that it only has a set up cost for the gateway and transceivers and in addition, very minimal maintenance costs, making it a highly desirable alternative to cellular, and accessible. We expect our project to cost ~150$ per unit (see the Bill of Materials), which is a fraction of the cost of even a single year of cellular making it good both in the short and long term.

The cellular system that is currently in place is not only expensive, but it suffers from having a lot of dead zones. Dead zones are very disruptive to the current system because when a bus is in a deadzone, it cannot connect to the main system, meaning that the bus can't be tracked via GPS. This is a problem because while fare collection data can be sent at the next connection site with few consequences to performance, this greatly impacts user experience as people are not able to track busses in real time until the bus is out of the dead zone. This is also an issue because TCAT can't use the data to better refine their systems for making efficient bus routes. LoRaWAN would not only provide us with a cost effective solution, but it would also greatly reduce the amount of dead zones that we are currently seeing. When we implement our system, we will not only be collecting fare data, but we will also be able to use GPS to track the busses, and the LoRaWAN system will allow us to connect to the central server more frequently.

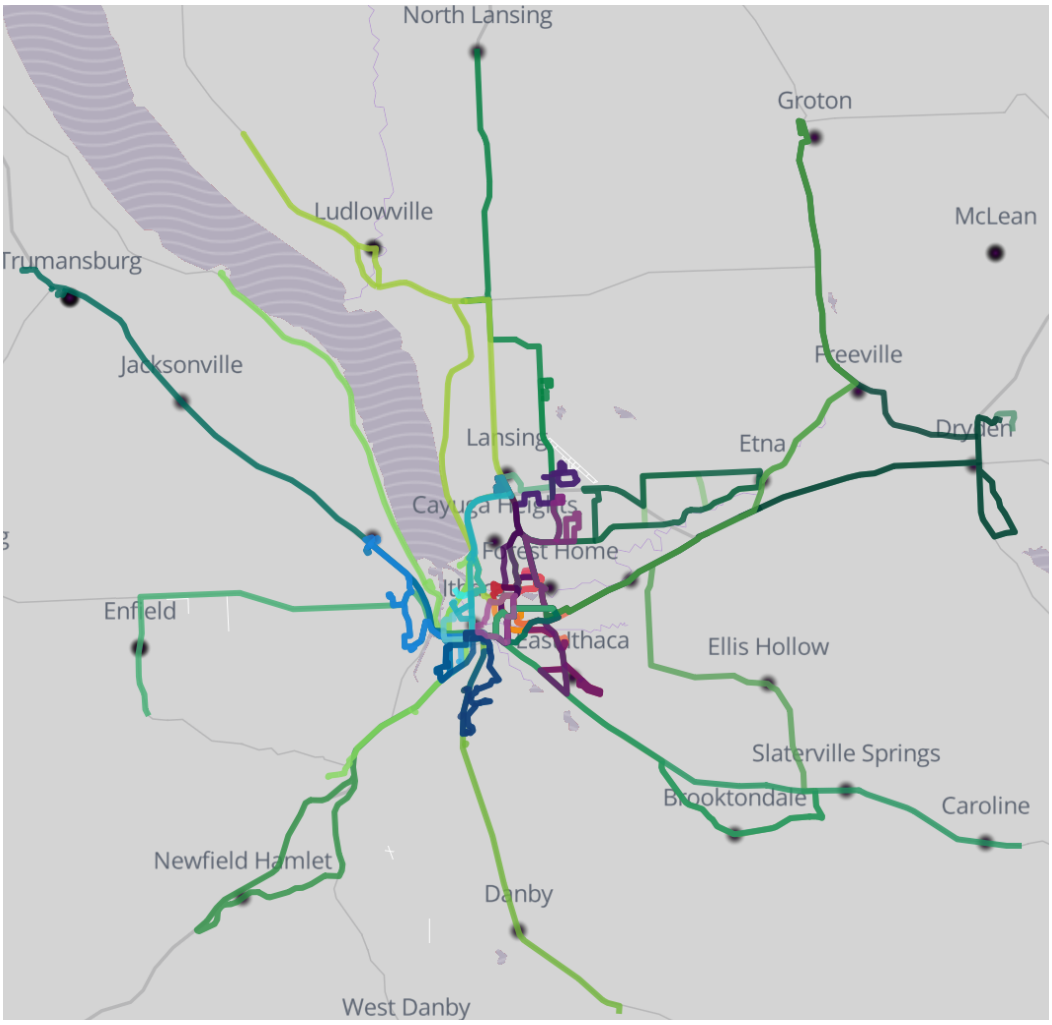


Figure 2: Route Map of TCAT Buses

Figure 3: Cellular Coverage of TCAT Buses

Our project will hopefully result in much better GPS tracking coverage which will allow for TCAT to collect much more and useful user data. TCAT values data transparency, while protecting the security and privacy of the customers who use it. The "Popufare effort is to provide as much data transparency as possible, while still providing a reasonable measure of privacy and security." [3] It is important for the TCAT bus system to collect detailed ridership data because it allows for them to plan group pricing for a group such as Cornell students, but also allows for analysis on bus route usage, and thus efficient and effective bus system operation. Our project would aid Popufare in becoming a more sustainable system.

In addition to all of the benefits our project will bring to TCAT, we also have a strong focus on accessibility, and as a result, we will ensure that our project is entirely open source. Our community partner describes, it "helps provide a model of offering free/libre software and hardware for public infrastructure that can hopefully be extended to other public infrastructure and services."[3] Not only does our project benefit TCAT specifically, but it allows for other smaller companies to afford the technology that they might not have been able to otherwise due to price. Being open source not only makes the technology accessible, but it allows for a company using it to be able to gain insight into the inner workings of the infrastructure they are using which is often hidden in premade plans. Our partner stated that if successful, our project may provide "a cheaper alternative to proprietary options as well as the potential for customization and specialization"[3] for companies. Since bus companies can deploy their own gateways to cover all the routes, our solution would be cheaper, more accessible, and more

robust. The overarching goal with our bus fare collection prototype is to use a LPWAN that might serve as a model for public infrastructure, and become more widely used in other applications, some that may have not been developed yet.

Review of Related Work

We are accomplishing the task of Asset Location Tracking (ALT), or Automatic Vehicle Location System (AVLS) in our case. There is a lot of market demand for ALT for big businesses have who have lots of assets from equipment to large parcels of inventory. Businesses risk a large financial loss if their assets are lost due to them being stolen, or misplaced. Some examples of where ALT is necessary include vehicle tracking, inventory tracking and management, and sometimes employee tracking. This data is helpful because it can not only be used for analysis but also for loss prevention and automatic inventory management.

There are many methods to do ALT. People often perform ALT themselves by manually updating spreadsheets, however this method is costly to scale and prone to human error. There are also automated ways to do this including and not limited to QR Codes, RFID, NFC, GPS, and Cellular Positioning to track locations of assets. [1] Each of these methods has its strengths and weaknesses such as being low power at the cost of speed or range.

Most places in the world have a form of GPS location services [23]. The part of GPS tracking that is difficult is sending GPS sensor data to the server to be recorded. Most ALT services use cellular or Wi-Fi connections [6]. Recently there has been a movement to start doing ALT using IoT devices [6]. This is because IoT devices usually do not require subscription services (other than an ISP), which is cheaper. IoT devices are always on, long-lasting, and allow for easy software updates. IoT devices can also be vetted for security and updated if a vulnerability is found [9].

We are specifically using the GPS method. This method works well for our application because the assets can be tracked from a very large area for low power. We are connecting IoT technology on the TCAT busses to LoRaWAN access points. LoRaWAN technology allows for long-distance connectivity for good coverage, full-duplex communication, and low power consumption [10]. This is why the industry already has a few LoRaWAN GPS asset tracking solutions. Several companies have also developed commercial products [4,8,11] that update their GPS coordinates over LoRaWAN. These devices are small, portable, and ultra-low power. They operate on power sources like AAA batteries with up to 5 years of battery life.

There are a few reasons why the industry has not yet adopted widespread IoT LoRaWAN ALT technology. LoRaWAN technology has only recently been accepted as a proven method for ALT. The industry is often slow to adopt large changes to infrastructure and there are still many other ALT methods that work just fine, making the demand to switch low. Lastly, LoRaWAN coverage is relatively low for certain asset tracking demands [2]. There are even alternative IoT methods that have the same advantages as LoRaWAN.

	Range	Coverage	Expense	Location Precision	Error Rate
Human	**10**	**9**	**1-2**	**10**	**6**
RFID	**1**	**1**	**8-10**	**10**	**10**
NFC	**1**	**1**	**8-10**	**10**	**10**
Cellular Positioning	**10**	**8-9**	**3-4**	**6-8**	**6-7**
GPS - LoRaWAN	**8-9**	**9-10**	**7-9**	**6-8**	**7-10**
GPS – Wi-Fi	**3-4**	**3-4**	**7-10**	**6-8**	**9-10**

Key: Range [1, 10]
1 = Worst Result
10 = Best Result
- All numbers are relative to each other
- X-Y is a result dependent on implementation from X to Y inclusive

Table 1. Comparing Implementations

Project Scope

The Vision. When meeting with our community partners and the teaching team at the beginning of the semester, we discussed two potential directions for the project: creating an API for Popufare to directly communicate through LoRaWAN without worrying about the implementation details, or plotting a coverage map of the gateways along bus routes. After further discussion among team members, we decided to break up the project into two stages and accomplish both directions. Ultimately, we are working towards the goal of replacing all communications between the units on buses and the central server to LoRaWAN entirely as opposed to a cellular network to provide better coverage when tracking the buses. However, we were not able to achieve this goal due to limitations in time.

The Design Objective. The first stage is implementing an initial demo with the Raspberry Pi, the LoRa transceiver, and the GPS module over TTN. Using our initial demo from stage 1, we can already plot a coverage map of the gateways as we would simply need to carry the device with us when travelling in Tompkins County. To make sure that we get a coverage map under realistic conditions, we modified our prototype for installation on existing TCAT buses. The second stage is developing the actual API that Popufare will interface with as well as our own “decoder”. Knowing that the parts are functional, we can build on top of our demo and implement functions that the system could call to transmit packages. We would also need to implement a decoder, likely a standalone program that may run on a separate device, for the central server. Its objective is to receive packets, reformat (perhaps reassemble) them, and transmit them to the actual TCAT server. Since we were not able to obtain access to the TCAT servers and we were short on time, we instead implemented a web interface using the Python library gmplot and a Google Maps API to plot coverage data automatically.

The IOT System Under Consideration. For our IoT system, we are using a Raspberry Pi 3 B as the main microcontroller. Connected to the Raspberry Pi are the GPS module and the LoRa transceiver. The Pi receives data from the GPS module and sends it through the transceiver. Using LoRaWAN and The Things Network, our server/decoder will receive the packets and decode it into the format that the current central server uses so that minimal modifications are needed for the actual system, as seen in figure 4.

For the Mechanical System Under Consideration For the mechanical system we built a 3d printed prototype box that would house the electronics. In addition we also built an antennae housing system that would protect the antennae against a powerful daily bus wash while also allowing it to function and have good access to the outdoors. We designed the box to be built for longevity.

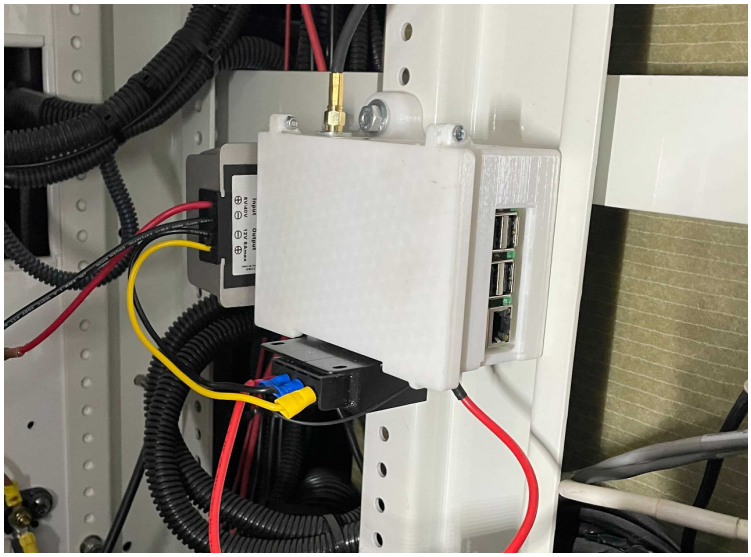


Photo of the Final Box Mounted onto the Bus

Photo of the Antenna Dome on Top of the Bus

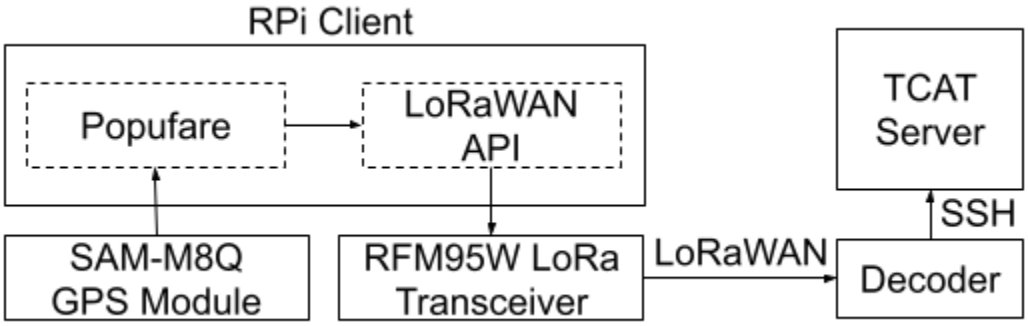


Figure 4. System Diagram for Vision

Technological Development and Results

The Mechanical System - The Prototype

Mechanically, the focus was mounting the box to not break and to last a long time. Originally the plan had been to secure everything to protect against vibration, however during the site visit we found that none of the electronic devices had any measures, so it is safe to assume that this wasn't an issue. We have our prototype built with each component secured with screws and anti vibration nuts so that it will not fall out during the tests. In the final prototype, we had more heat generating components such as the pi which we put on stilts, the buck booster, and the step down converter which are both mounted further away from the box on the outside and are on stilts. The box also comes with a hinged lid for access to the components and a lock to keep the lid closed. The box also is mainly hung by a thick loop on the top of the box that is close to the box to ensure that it is stable, and on the opposite side on the bottom the box has another loop for securing it down there as well. The pi's ports, visible on the right hand side of the image, are open here, however a door exists that can slide into place with a friction fit. This will eventually be changed to slide in from the other side so that gravity will keep it in place. A 3d render of the cad is available on appendix A.

The Electrical System

Since the ultimate goal of the project is to modify an existing system with existing hardware, the project falls more onto the software side with the mere addition of a LoRa transceiver. However, for testing purposes, we also included a GPS module so we can transmit actual data when conducting field tests.

For the GPS module, we selected the SparkFun GPS Breakout for SAM-M8Q (u-blox). GPS, which stands for Global Positioning System, was developed by the US military in the 1970s. Its goal is to provide geolocation and time information to devices anywhere in the world based on the simple principle of triangulation. For any point in 2-d space, if its distances to three known locations are known, its coordinates can be determined. GPS uses a similar but more comprehensive technique called multilateration that involves clock offsets. There are currently 32 satellites revolving around Earth whose locations are known and clocks synchronized to each other and the ground. Any GPS receivers that can receive signals from at least four satellites can

calculate their current location and calibrate their onboard clocks based on the time offset between different signals.

Because we need real-time locations of buses, we need a GPS module. We picked this breakout board because it is great for prototyping. It uses qwiic connectors, which eliminates soldering, and it has a chip antenna, which means there is no need to add an extra antenna to our system. In addition, it is a 72-channel GNSS receiver that receives signals from the GPS, GLONASS, and Galileo constellations, which "increases precision and decreases lock time." It also contains an onboard rechargeable battery that powers the real time clock, allowing it to get a hot lock in much shorter time. Overall, the module has great performance and is very easy to use. For the actual Popufare system, however, it might be preferable to select a module that supports an external antenna for better GPS signal. We do not have sufficient data for the performance of this particular module on TCAT buses.

For the LoRa transceiver, we selected the Adafruit RFM95W (Semtech) LoRa Radio Transceiver Breakout with DIYmalls 915MHz LoRa Antenna from Amazon. We picked these because they have a great range of about "2 Km line of sight using simple wire antennas, or up to 20Km with directional antennas and settings tweakings." It also uses SPI as the communication protocol, which makes the library easy, and has the correct modulation frequency. Since range is very important for our application, we opted for an actual LoRa antenna (omnidirectional) instead of using wire antennas. The antennas come with short U.FL/IPEX to SMA pigtail cables that are approximately 10 centimeters long. When installing our prototype on TCAT buses, we want to maximize the range and minimize the interference and obstructions of the signal. Therefore, we made our own SMA to SMA extension cable using 20 AWG 50Ω coaxial cable. The extension cable is about 3 meters long and should be able to reach the roof of the buses, on which we intend to install our antenna.

Buses provide onboard power, so we can use that to power our device and not worry about power consumption. Our device would draw power no larger than 10 watts (very much an overestimation), which is miniscule compared to other electronic components running on the buses, such as headlights or internal lighting. We are using an 8V-40V to 12V buck-boost converter in combination with a 12V/24V to 5V converter that directly connects to the 5V pin on the Raspberry Pi.

A circuit diagram for the connections between modules can be found in appendix B.

The Software System

Demo Client

As seen in figure 5, our demo program is very straightforward. Our original demo program (in Python) collects the GPS measurements, including the latitude, longitude, and time, and sends the data along with the bus ID through TTN. We later added the ground speed, the speed accuracy, as well as the motion heading to the packet, which now has the size of 28 bytes. It is very simple to implement functions in C that interact with the Python source code through embedding [15] and using *PyByteArrayObject* [16] or a char array (string) to pass data between Python and C [17].

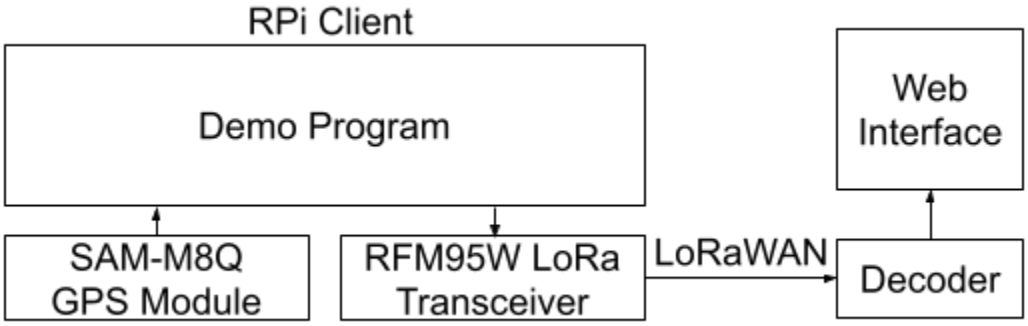


Figure 5. System Diagram for Final Prototype

Receiver & Output System

While conducting laboratory and field testing, we updated our receiver and output module to have our output graph on a web interface using a Google Maps API rather than a static image. Our system evolved into the following structure accordingly:

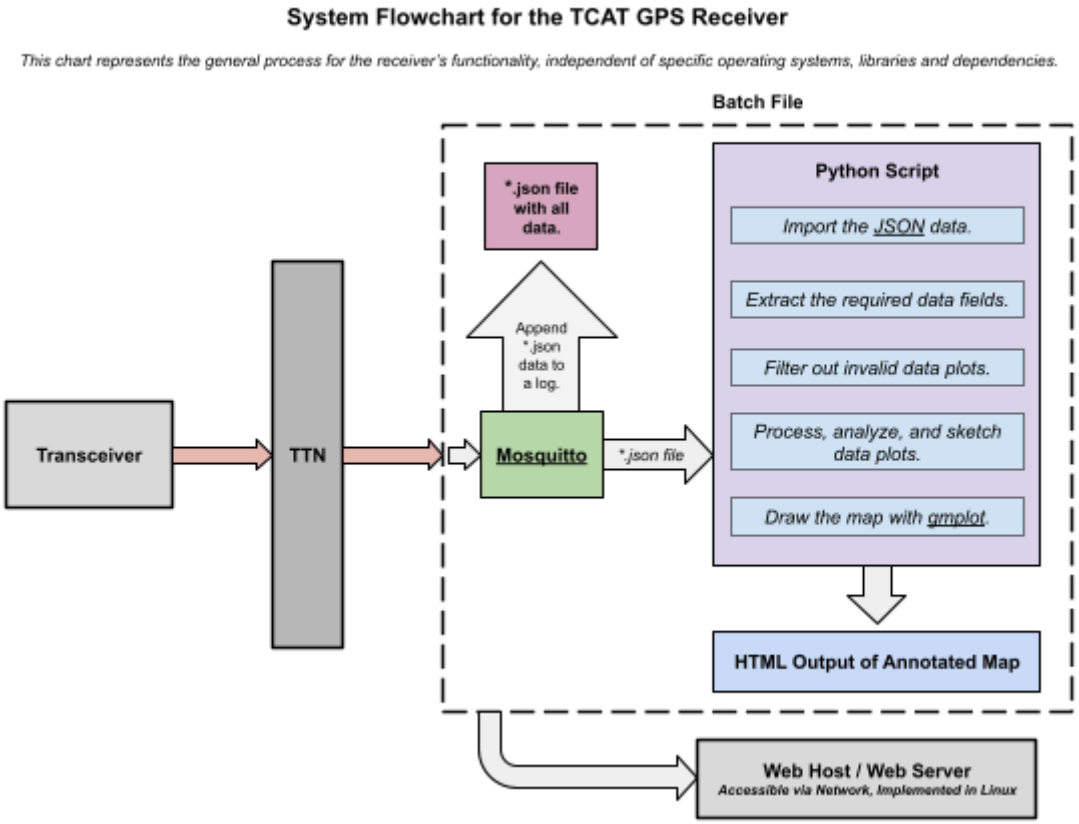


Figure 6: System Flowchart for the GPS Receiver's functionality.

In the system's current form, the Linux implementation of the receiver has 3 main parts to dictate the functionality of the program. The first part is a command line menu that prompts the user to either load in fresh data from the TTN console to map out, import a .json file from previous data collection, or quit out of the menu. New data gets backed up to a json log file. Once the user chooses, they will navigate to the main part that allows them to load in the data. Regardless of the source of data that the user chooses, the other 2 main parts process that data in the same fashion. Using a python script that employs the Google Mapper Plot, or gmplot library, the program will extract the longitude and latitude, as well as the timestamp, velocity and device id, process and clean the data so that only valid coordinates are shown, and then visually map the resulting progression of the GPS points in time on a dynamic map.

Results from the Map

As previously mentioned, many aspects of the receiver for the GPS have undergone drastic changes throughout the development of this project.With these changes came adjustments to the appearance of the map's interface for the data.

In the first rounds of testing, rather than getting the mapped out data as an HTML output, we used Python's GeoPandas to output a static image of a map with the GPS coordinates plotted.

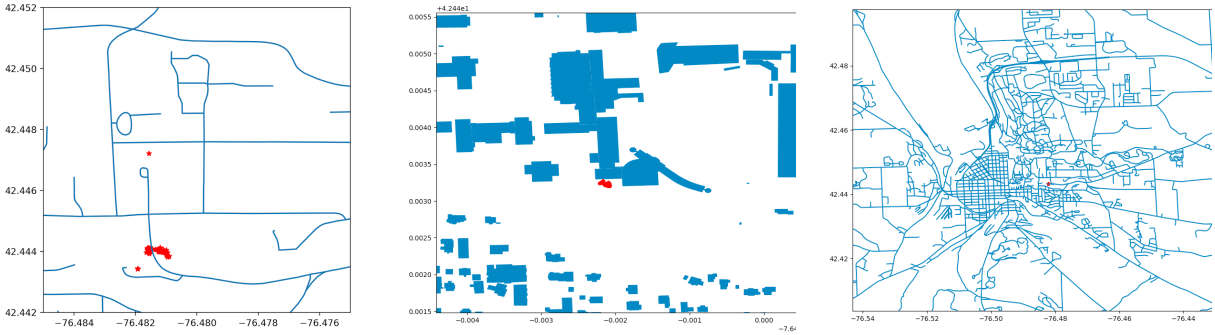


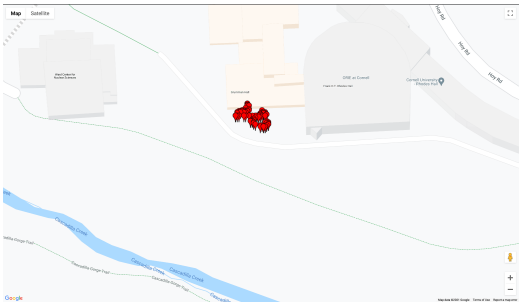
Figure 7: Original testing data.

These images work well for laboratory testing because it's simple, read-only data that only maps out areas with data. The *GeoPandas* maps show enough of Ithaca's shape that we can recognize the features of the areas in which certain points are plotted.

However, this approach was not feasible for plotting our data because it does not allow us to easily recognize the geographical features of or explore data from regions that we are not as familiar with. Additionally, *GeoPandas* was somewhat limited in ways to present the data. So, for our field testing, we switched our interface to show a more dynamic map with our data that would allow us to present our data in a more user-friendly format. This requires a web server to host the .html file output. The results of live data can be viewed via this link, the results of pre-loaded data via this link, or in the screenshots below.

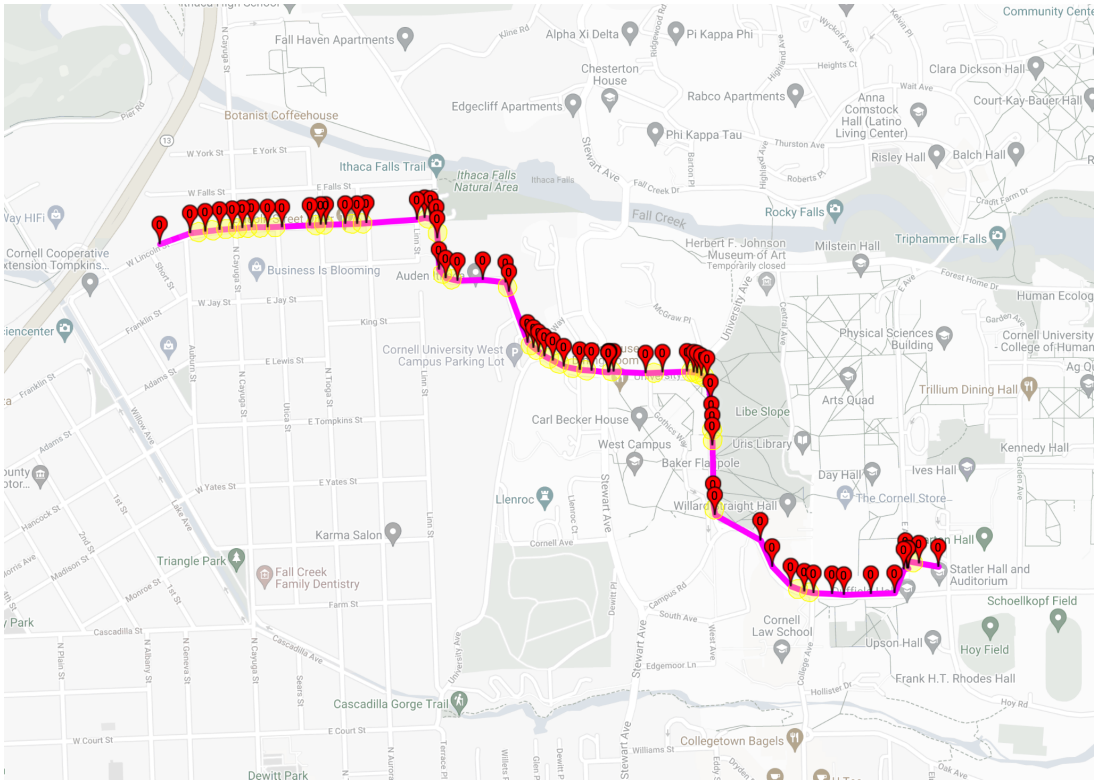
This interface is set up just that one can either upload a *.json file to view pre existing data, or one can pull the newest data from the TTN console and plot it. The new data from TTN

is always saved to a backup file to revisit later. On the map, the course of each GPS is mapped with marker data points as well as a color coded trendline. We can infer which points are missing from the data collection by looking at the areas with the most red markers and at the fit of the trendline relative to the roads. However, this does not indicate whether certain areas lack density in data points due to coverage issues or due to the speed at which the vehicle is travelling. For this reason, we are using the velocity calculated by the GPS, the data counter/index, and the intervals from time stamps to predict and plot points that may be missing due to speed, in the color yellow.

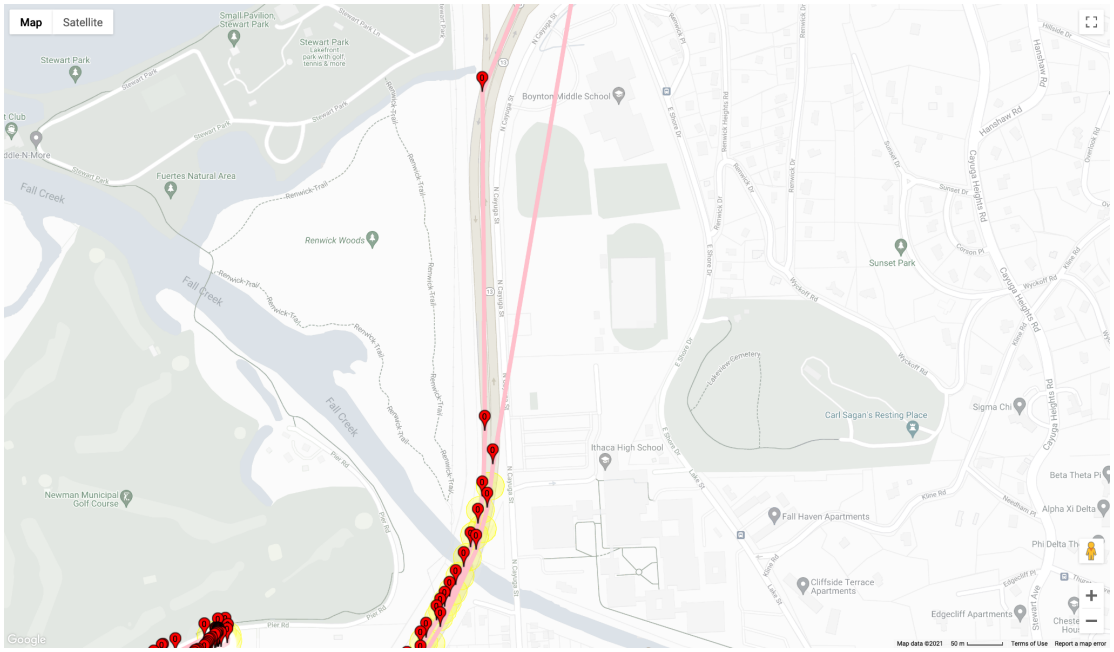


The map view of the data (test run 1).

The satellite view of the data (test run 1).



The map view of the data (field test 2).



This is an example of results showing areas of coverage and lack thereof in an area where a bus would be travelling at a high speed. Here, there seems to be a lack of coverage in the area of the highway overlooking Stewart Park and the main body of Cayuga Lake (field test 1).

Figure 8: Field testing and data collection.

From the data we have collected, we can see that most common areas where there are dead zones are near cliffs, large bodies of water, gorges and other prominent land gaps. Given the opportunity to continue this data collection and comparison over an extended period of time, we would continue to see this trend, which indicates that there is an opportunity to pursue ways to improve the coverage in these geographical areas by devising environmentally conscious methods to embed more gateways in these areas to increase the amount of data sample one can get. Overall though, these dead zones seem to be limited to the area immediately adjacent to these land gaps, and the number of data points we can detect per minute on average is 6-10 with LPWAN (TCAT-IOT Team).

Plan for the Next Steps

Further Research and Development. There are many aspects of our project where further research and development may be beneficial, and even essential to the reliability and potential for expansion of this system.

In terms of the electrical system, an effective way to improve upon the model is to use HATs for the Pi and solder/hot glue connections to different modules to ensure a stable connection. During testing, we saw some errors in the data received. We do not believe that separate modules or even the wireless communication is responsible for this. It is more likely that the vibration is leading to signal integrity degradation because our prototypes simply have jumper wires plugged into each other.

In terms of the software system, several parts still need to be developed. For example, embedding Python methods that we wrote in C and actually integrating these modules into Popufare. On the receiver end, the device also needs to be able to replicate the communication process currently implemented on Popufare.

To improve upon the current interface for receiving and displaying the GPS data, there are a couple of systematic and cosmetic changes that could be made. In terms of the system itself, a example of an effective development would be to set up the server that will power/run the receiver application to also act as a web server so that the interface that shows the GPS data on the map can be available to view by all employees on the internal office network / VPN as a web application. This would not only make the data more accessible to those who are meant to be authorized to view and access the application, but this move would also allow TCAT to protect the integrity of this data using the same protocols and under the same system as other internal applications and documents. Another benefit to this proposed setup would be that TCAT could later choose to attach a domain and make the interface available while allowing the server to remain guarded within the internal network's firewall and security settings. This would be a major upgrade from the Github-based system for updating and hosting the interface. Using a given operating system's task scheduler to load in the new data would allow this server to run autonomously for the most part, with a few periodic maintenance check-ins. At the most fundamental level, this can be accomplished by setting up a Raspberry Pi, as was done for lab and field testing. The Raspberry Pi could be connected to the router of the main internal network via ethernet cable and thus be accessible as an internal server that a user can ssh into.

Another opportunity for research and development in terms of the software interface would be to look into different options for mapping the data that are also operating-system independent, easy to install & implement, and are sustainable as well as customizable. An important part of this project has been the commitment to creating an open-source solution. The current system uses the *Mosquitto* library to capture data from TTN in a JSON-like file, as well as the *gmplot* library to display the data from the GPS on the map. We previously used the *geopandas* library for the data processing and display, but we made the change to create a more dynamic and user friendly interface for the results, and to allow the application to be run more efficiently on a Raspberry Pi and other Linux-based systems without impacting its usability on other operating systems. While this library enables a clearer representation of all the data with an option to label, differentiate and predict the trends of certain points, and it makes the application easier to deploy, there are tradeoffs. The *gmplot* library alone does not have as much data handling capabilities in its own right, and using certain features that could improve the function of the interface in the data analysis realm requires the purchase or trial of a Google Maps API key. Depending on the needs of the system as the project evolves and what can be compromised, this might be an aspect to look into more. Finally, the interface can be improved by being adapted to read in larger *.json files (greater than 300KB) more effectively.

Mechanically, the majority of what will come next will depend on what new features are added to the project if any. The box itself will grow to accommodate any new items or change to the wishes of the TCAT team to better fit. In addition, currently the box and antennae set are in testing on the busses themselves, and so we will be able to see how well the antennae's shield

withstands the force of the bus cleaners which work every morning, and in addition, we will have to see if the thickness of the shield blocks the signal noticeably and if so, make proper adjustments.

Barriers to Development. There are some existing barriers that directly stem from the social context and from the environment for which this system is being developed. For instance, the degree of existing coverage of the new system is a resource-based barrier that will require there to be more gateways installed as more dead zones are discovered. In that sense, our project by nature exists to help break down that barrier.

Another barrier to development may be the fact that TTN is a public LoRaWAN that is community-based. Since every device has a chance to use it, TTN "limits the uplink airtime to 30 seconds per day (24 hours) per node" [5] so that every device has a chance to transmit. Each of our packets, as bare bone as it is, has an estimated airtime of about 70ms. Currently, Popufare has an update frequency of once every fifteen seconds when the bus is in motion and once every sixty seconds when the bus is still. Assuming that each device is only turned on for eight hours per day, which is very likely an underestimation, and that a bus updates its location approximately every forty-five seconds, the uplink airtime per day would be about $\frac{8\times3600}{45} \times 70ms = 44.8s$, way over the daily limit. To mitigate this issue, it might be preferable to construct a private LoRaWAN. Alternatively, LoRaWAN could act as a backup or supplement to cellular network if there is no reception. This will only solve the coverage issue but not bring down the cost of the system.

Funding Mechanisms. In Dr. Wessel's two-part lecture on social entrepreneurship, we reflected on the Revenue/Business model of the TCAT Fare System with respect to a transition from Cellular coverage to LoRaWAN coverage. The key costs of implementing our LoRaWAN based system across the entire local transportation system are as follows:

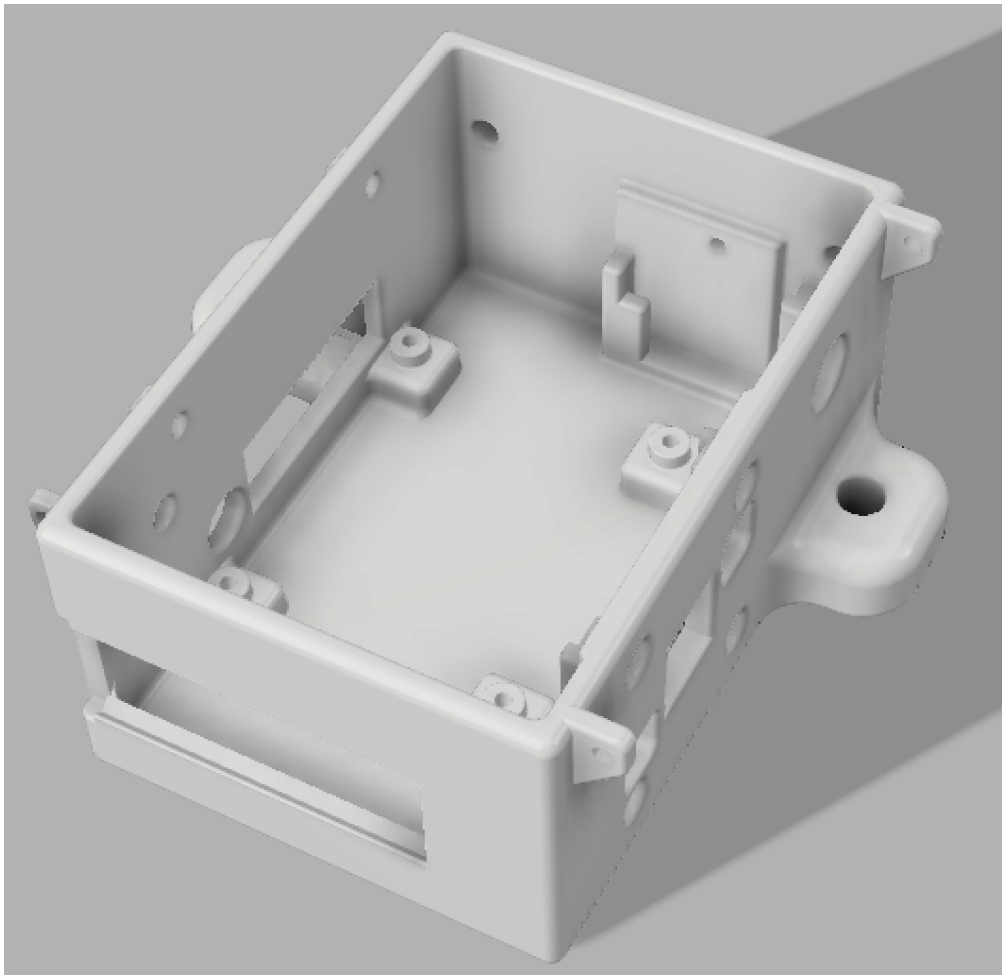
- The cost to engineer novel system (start up)
    - Buying the equipment
    - Integration
    - Manufacturing
    - Design
    - Installation
- The cost to maintain system (fixed)
- The cost to repair (variable)
- The cost of updating / innovating the system - including installing more gateways to improve coverage (variable)
- The cost to produce additional units for the system for expansion (variable)
- The cost of having insurance on the units (fixed rate, varies by number of units)

With respect to the cost to engineer and implement the system, a potential recurring cost might be obtaining an API Key from Google Maps for the mapping and satellite information for the streets and other geographical features of the surrounding area. Additionally, due to the fact that this system is meant to replace the existing cellular system, current employees can be retrained in managing the new system, so no additional cost for new staff is needed.

In addition to the local transportation industry and residents in Tompkins County, this product has the potential to serve rural communities who currently have less reliable transportation systems across the United States and the E.U. For instance, the Montgomery County Transportation System is of a similar scale as TCAT, with comparable hardware and software implementations and issues. According to the Montgomery County Department of Transportation, traffic congestion is a huge barrier to reliable transportation. This county, like others, could find great utility in using this project's data to analyze traffic patterns over the course of a day in each of the county's towns. Using this data, the county could initiate public good projects to improve traffic flow in the county, which is a major value proposition that would improve transportation reliability and the quality of life in that county as a whole.

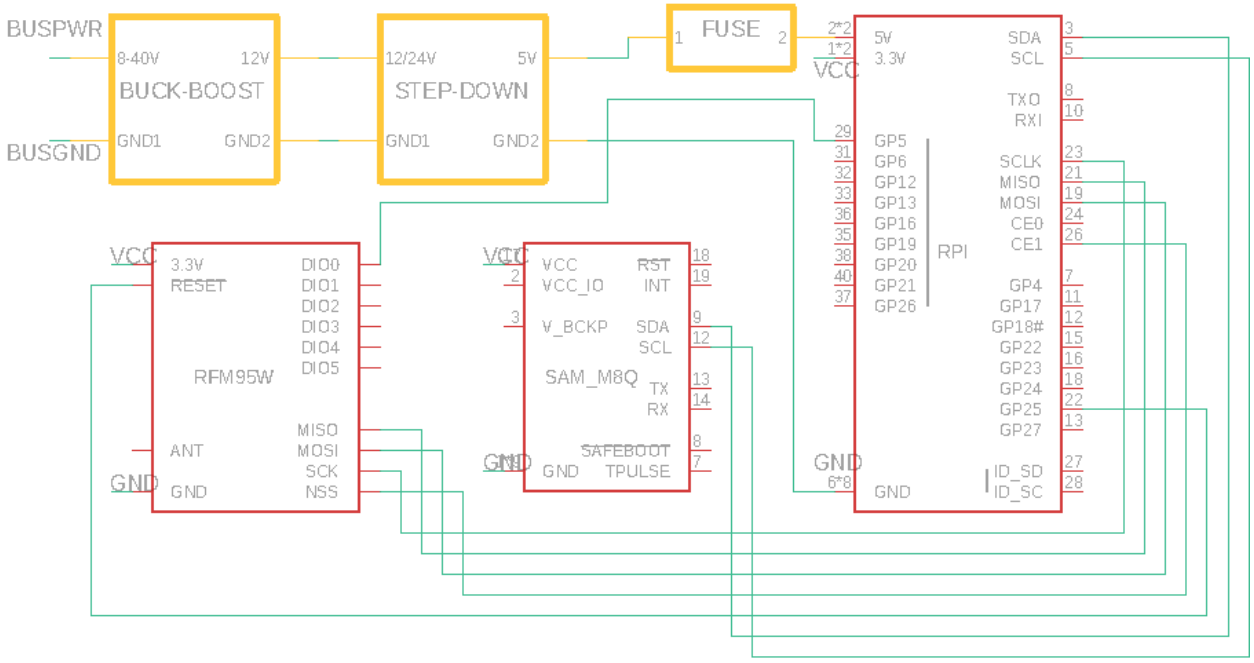
In fact, the National Transportation Safety Board and/or the U.S. Department of Transportation could pay for this kind of data on a regular subscription basis to research and analyze the current state of traffic flow. This benefits them by allowing them to better strategize in advising states and counties in their infrastructure and regional planning as needed, and allows them to make more informed decisions on how to allocate federal funding to combat the issues uncovered by the data. Other value propositions of this project include, but are not limited to: increased understanding of vehicle traffic properties, decreased BUS tracking costs, and better bus tracking for riders.

Appendix A. 3D Models



3D Render of the Final Box

Appendix B. Circuit Diagram and Source Code for Final Demo



Our source code can be found at https://github.coecis.cornell.edu/mdd94/TCAT-IOT.

References

1. *7 Most Common Methods Used for Asset Tracking - Asset Infinity.* Asset Infinity Blog. (2019, November 5). https://www.assetinfinity.com/blog/asset-tracking-methods.
2. Actility. (2019, January 18). *The Business Case for Densifying LoRaWAN Deployments*. IoT For All. https://www.iotforall.com/business-case-densifying-lorawan-deployments.
3. *Connelly, A. (2021, April 19). Update for IOT TCAT Project and Questions.*
4. *Dragino LoRaWAN GPS Tracker with 9-axis IMU-LGT92 - 915MHz*. Dragino LoRaWAN GPS Tracker with 9-Axis IMU-LGT92 - 915Mhz - Seeed Studio. (n.d.). https://www.seeedstudio.com/Dragino-LoRaWAN-GPS-Tracker-with-9-axis-IMU-LGT92-915Mhz-p-4536.html.
5. *Duty Cycle*. The Things Network. (n.d.). https://www.thethingsnetwork.org/docs/lorawan/duty-cycle/.
6. EMnify. (2020, August 5). *IoT Device Security: Comparing Cellular, LoRa, & WiFi*. IoT For All. https://www.iotforall.com/iot-device-security-comparing-cellular-lora-wifi.
7. *gmplot. PyPI. (n.d.). https://pypi.org/project/gmplot/.*
8. *GPS Industrial Tracker*. LoRa Alliance®. (2020, November 24). https://lora-alliance.org/lora_products/gps-industrial-tracker/.
9. *IoT-enabled asset tracking*. Thales Group. (n.d.). https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/industries/asset-tracking.

10. Lora Alliance. (2020, April). *WHY LoRaWAN® IS THE LOGICAL CHOICE FOR ASSET-TRACKING CONNECTIVITY*. https://lora-alliance.org/wp-content/uploads/2020/11/WhitePaper_AssetTracking.pdf.
11. *LoRaWAN® GPS Trackers: Low-Power Devices, Global Support*. Digital Matter. (2021, May 6). https://www.digitalmatter.com/our-devices/lorawan-gps-trackers/.
12. *LORAWAN GPS Tracker*. IOT Factory. (2020, November 10). https://iotfactory.eu/products/iot-sensors/lorawan-gps-tracker/.
13. *MoCo's Traffic Congestion is Getting Worse*. Empower Montgomery. (n.d.). https://www.empowermontgomery.com/mocos-traffic-congestion-is-getting-worse/.
14. *National Transportation Safety Board*. USAGov. (n.d.). https://www.usa.gov/federal-agencies/national-transportation-safety-board.
15. Python Software Foundation. (n.d.). *1. Embedding Python in Another Application*. 1. Embedding Python in Another Application - Python 3.9.5 documentation. https://docs.python.org/3/extending/embedding.html.
16. Python Software Foundation. (n.d.). *Byte Array Objects*. Byte Array Objects - Python 3.9.5 documentation. https://docs.python.org/3/c-api/bytearray.html.
17. Python Software Foundation. (n.d.). *Python/C API Reference Manual*. Python/C API Reference Manual - Python 3.9.5 documentation. https://docs.python.org/3/c-api/.
18. Sajingeo, & Instructables. (2017, October 26). *Host Your Website on Raspberry Pi*. Instructables. https://www.instructables.com/Host-your-website-on-Raspberry-pi/.
19. TCAT. (2021, February 25). *FAQ*. TCAT. https://tcatbus.com/about/faq/.
20. TCAT. (2021, March 13). *Ridership*. TCAT. https://tcatbus.com/about/ridership-and-statistics/riders/.
21. TCAT-IOT Team, Cornell University, Spring 2021. (n.d.). *TCAT Coverage Map Data*. Google Maps - gmplot. https://pages.github.coecis.cornell.edu/mdd94/TCAT-IOT/.
22. U.S. Department of Transportation. (2021, May 14). U.S. Department of Transportation. https://www.transportation.gov/.
23. *What is a GPS? How does it work?* The Library of Congress. (2019, November 19). https://www.loc.gov/everyday-mysteries/item/what-is-gps-how-does-it-work/.
24. *Working With Real Data and Managing Power*. training-labs. (n.d.). https://pages.github.coecis.cornell.edu/LPWAN-Training/training-labs/lab2.