Jarod Lopez

Dr. Galathara Kahanda

Computer Science Project

December 10, 2025

Lunar Internet of Things Final Report

An Internet of Things (IoT) is a system of devices, embedded with softwares and sensors allowing them to communicate through shared network connectivity [1]. IoT systems can be applied in many different problems, one of which is the lunar surface traversal. With Dr. John Schmalzel of Rowan University, our team recreated the lunar surface and constructed an IoT within it. Our IoT consists of a command center, rover, battery stations, and objectives. The rover must move between objectives and battery stations, ordered by the command station, in the most efficient path possible to preserve power levels. When significant battery power is lost, it must request battery information from battery stations and determine where to recharge. The physical objects are also going to be recreated as digital twins, virtual objects or systems that use real-time data to reflect changes made to their physical counterparts, in an online environment using Unreal Engine [2]. This project is inspired by the Lunar Command and Control Interoperability (LuCCI) project by NASA. The LuCCI project aims to address how sensors and other devices on the lunar surface would interact with one another and send each other necessary data.

This project required many different hardware and software materials. The full supplies list for this project included:

- Hardware: Raspberry pi, MicroROS rover, Power banks, Memory cards, LEDs, resistors, jumper wires, Servo motors, ESP32 development board, Cameras

- Software: Space ROS, Unreal Engine, Unity, Autodesk Fusion360, Cesium, OpenCV, MQTT Servers/Brokers

The group (Julie, Sam, Emily, Lekh & I) have met every week to work on this project. The group's roles are as follows:

- Julie is responsible for ROS2, pathfinding, mock lunar surface design, and digital twin development.
- Samhith is responsible for ROS2, pathfinding, and battery/robot modeling simulations.
- Emily (PM) is responsible for working on battery/robot modeling simulations, mock lunar surface design, and digital twin development.
- Lekh is responsible for working on communication over an MQTT broker, as well as guiding the team to successfully recreate his lunar environment and add onto it.
- I (PM) am responsible for working on computer vision components and communication over an MQTT broker.

Regarding computer vision, our original plan was to use the overhead ESP32 camera to split the lunar surface into a coordinate grid that we could project a weighted map onto. However, as we developed this program, we realized that it would be difficult to do this without extensive machine learning algorithms. It would also create too heavy of a workload to continuously send images (which come from the video stream) over an MQTT broker. Our initial solution to this issue was to implement TinyML, machine learning models that are designed specifically for low-power machines, such as microcontrollers. One tool that we had attempted to use was Edge Impulse, an AI platform that simplifies building and deploying TinyML models. To use their YOLO model, which creates a bounding box around identified objects, we would need to train the model using images taken with the ESP32 camera of the objects to be identified. However, after training with numerous images (around 50 for each station/objective), we were only able to achieve an accuracy rate of around 50%. This would be too low for the goals of the program and could cause significant issues with rover pathing, so we decided to consider a different solution. Since each objective and battery station is a different color, the new solution we created was to upload our color detection code to a Raspberry Pi and create bounding boxes around the identified colors in the video stream of the overhead ESP32. We would end up optimizing this code for nearly a week as we had to make sure the HSV value ranges of the program could correctly identify the colors of our 3D models in our testing environment, but it successfully tracks each model. The rover is identified with a color range encompassing different HSV values of black as most of the rover body is black.

With MQTT server communication, I was in charge of creating a program that would allow the ESP32 microcontrollers of the battery stations to subscribe to a topic on an MQTT topic and listen for information requests. Once a request is received, the microcontroller would then send the battery information of the power station it is attached to to another MQTT topic for the rover to receive and utilize. Having little experience coding in C++ and using Arduino IDE, I had taken some time to complete this task as I had to look into example codes and beginner tutorials for programming with ESP32 microcontrollers. However, even with this setback, I was able to create a program that allows the ESP32 to connect to wifi and send and receive messages from an MQTT broker. Another aspect of MQTT communication that I worked on was the sending of the coordinates of the bounding boxes of the colors the Raspberry Pi identified. In Python, we used a class called paho-mqtt to establish connection to an MQTT server and send the box coordinates to the rover once every two seconds. With these coordinates, the rover could figure out its position in our mock lunar surface and find optimal paths to the stations and objectives.

Another responsibility of mine was to help piece together the physical objectives and rover stations. Lekh had created 3D models of the stations using Tinkercad, a free web modeling application, and printed them using the 3D printer at the Rutgers Dana Library. Emily and I were in charge of decorating the 3D prints, and we decided to paint them the same colors as the LED lights that would be attached to them (red, blue, yellow, and green). After decorating the stations, Lekh and I attached the ESP32 and other electronic components to the stations using hot glue. Lekh and I were successfully able to assemble stations and objectives that light up to allow for color identification with our computer vision program.

We tested each person's individual components in isolation as we completed them. Once we ensured that everyone had functional parts (i.e. accurate color detection, correct pathing algorithm, moving models in Unreal), we then worked together to update each component to work alongside others. Once everything was connected, we conducted a full test of the system to ensure success. Together, we were able to create a fully functioning program that allows a MicroROS rover to travel distances between objectives on the lunar surface and a virtual replica that mirrors the physical counterpart.

Although we did successfully complete the major part of the project, there were some parts we were unable to work on due to either time constraints or resource constraints. For one, we were unable to integrate the digital twin into Dreamscape, a virtual reality gaming platform. This could be attributed to the extensive time that it took to even get the 3D model of the rover moving in a virtual environment, but also to an overall lack of experience and resources in Dreamscape. Samhith and Emily had trouble throughout the semester getting the rover virtual model to interact with the virtual replica of the moon. For example, the mesh grids were preventing the objects from making contact, so the rover model would fall through the moon and out of the simulation. After too much trouble, Samhith eventually decided to move to Unreal Engine as it proved to be more reliable and easy to work with. This move is what enabled the digital twin to work correctly. Also, outside of the professors and graduate students from Rowan University who conduct research in Dreamscape, there were not many tools online to help us understand how to upload our virtual environment into the virtual reality world. Another goal we did not complete was to create a UI for users to easily send commands to the rover from and see the visual components of the project update in real time. We can attribute this to delays caused by underestimations of how long it would take for us to complete the individual components of this project. If we were given more time with the project, I believe we would have been able to successfully integrate the digital twin into a virtual reality environment and create a good-looking UI for rover controls.

Overall, I believe that this project and course provided a really helpful environment for us to test the skills we have been learning and building throughout the computer science curriculum at Rutgers. We were able to employ languages and software engineering frameworks that we were taught in earlier classes. This course also gave us the unique opportunity to work with outside mentors, specifically one who has experience working from NASA, so the project gave incredible insight on what we would need to consider when working on projects in the actual workforce. It also taught me new skills, such as microcontroller coding and computer vision, which I had not expected to learn before the end of my career at Rutgers. These are skills and knowledge that I will take with me past my time at Rutgers and will make sure to incorporate in my own personal projects.

Journals

October 10, 2025

In-person meeting at library. Lekh created a list of items to be ordered for the project. Sam and Emily worked on getting the basics of Unity down and looked into how to upload the rover 3D model to the digital environment. Lekh and Julie worked on setting up the new Raspberry Pi to allow for ROS commands but experienced some difficulties getting it to work. I worked on a computer vision program that identifies colors using HSV ranges. This program will be uploaded to rover to allow it to identify battery stations and objectives marked by colored LED lights.

October 17, 2025

In-person meeting at library. Materials were received from Amazon, so most of our time was spent putting things together. I assembled the robot with the camera and wired its components together. Sam and Emily worked on adding the rover to the Unity environment. Julie and Lekh worked on getting the newly assembled rover to begin taking commands, as well as finish setting up the Raspberry Pi. Met with Dr. Schmalzel to discuss midterm presentation and received notes as to what should be discussed in our presentation.

October 24, 2025

In-person meeting at conference room. Sam worked from home. Held a Zoom meeting with Dr. Schmalzel and his graduate student to learn about a graduate student's work in uploading a rover digital twin in Unreal Engine. Emily asked questions about uploading the lunar terrain into Unity as there have been some issues. I worked on completing a computer vision program to split the table surface into a coordinate grid without warping via perspective transform. Lekh and Julie worked on rover commands and setup.

October 31, 2025

The group worked from home. We held a Zoom meeting to discuss who would work on what moving forward. I will work on overhead cameras, MQTT integration, and battery stations. Lekh will work on MQTT, circuits, and 3D printing. Emily will work on rover movement in Unity and integration of MQTT in Unity. Sam will work on Unity and a cost function for our lunar surface grid. Julie will work on PyBaMm and the Raspberry Pi-ROS virtual machine.

November 7, 2025

Group visit to Rowan University with Dr. Kahanda. Held a meeting and tour with faculty from Dreamscape Lab. Sam and Emily got help getting Unity to load the lunar terrain faster and received tips for uploading digital twins to Dreamscape environment. Lekh and Julie connected the rover to WiFi and established basic movement commands. I worked on getting ESP32 to send battery info to MQTT broker topic as a JSON string.

November 14, 2025

In-person meeting at conference room. Worked on assembling battery stations (painting and gluing together 3D printed parts). Sam and Emily worked on rover motion in Unity. I completed MQTT broker communication between ESP32 battery stations and rover requests. Julie worked from home and installed PyBaMm on her computer.

November 21, 2025

In-person meeting at conference room. Battery stations were assembled and painted, so LED lights were added during this meeting. Began looking into Edge Impulse as a TinyML solution to overhead cameras. Trained an example model to identify pen and fork with ESP32 camera. Sam and Emily worked on Unity and Unreal Engine to complete digital twin and lunar surface simulation. Julie worked on battery modeling and rover movement.

December 5, 2025

In-person group meeting at conference room. The overhead tripod arrived, so we were able to finish work with the overhead camera. Edge Impulse TinyML model was not working as accurately as intended so this week was focused on getting the overhead ESP32 camera to identify battery stations, objectives, and rover by color, get bounding box coordinates, and send them to the MQTT broker. Successfully completed asset tracking with cameras. Emily and Julie worked on assembling the lunar surface for our demonstration (i.e. printing and piecing together tiles of the lunar surface). Sam worked on final touches in Unreal Engine (after deciding to make a late switch from Unity) and Lekh worked on rover and custom Wi-Fi server using Raspberry Pi.

December 9, 2025

In-person group meeting at conference room. This meeting was mostly spent testing individual components, such as pathing algorithms and sending of objective coordinates over the MQTT broker, and piecing them together. Successfully, our group was able to get everything working together and have the rover move between objectives, taking into account distance and battery levels.

References

1. IBM. “What is the Internet of Things (IoT)?” *IBM*, 17 Oct. 2025, www.ibm.com/think/topics/internet-of-things
2. IBM. “What Is a Digital Twin?” *IBM*, 17 Oct. 2025, www.ibm.com/think/topics/digital-twin