Remote Beehive Monitoring Utilizing LoRa Radio Technology

Gavin Cianciolo, Tyler Fitzpatrick
Innovation Scholars – Internet engineering Cohort, University of New Hampshire, Durham, NH 03824

University of New Hampshire InterOperability Laboratory

NEW HAMPSHIRE BEEKEEPERS ASSOCIATION

Introduction

Bees are the primary pollinator of flowering plants including many staple food crops and are vital to our survival. Keeping bees in colder climate regions such as New England presents the challenge of successfully overwintering a hive. By providing beekeepers with a means to collect meaningful data about the internal climate conditions of a beehive, not only during the winter when physically inspecting the hive is impossible but also during the rest of the year, they can have a better understanding of the health of their hive and better prepare for or react to critical conditions or situations. By making this system capable of being utilized in remote areas, its usefulness increases drastically, as it can be used to monitor large-scale apiaries (collections of beehives) such as on farms or orchards where Wi-Fi and electricity are scarce or impossible to access.

Project Goal

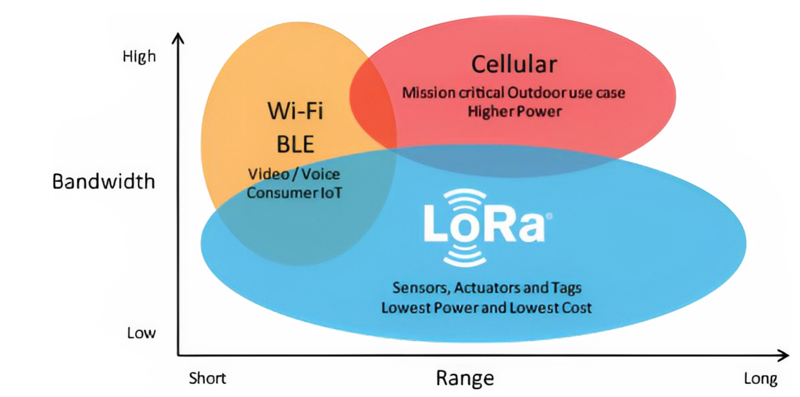
Develop a system that can remotely monitor the conditions within a beehive that are vital to the colony's survival, which can be used in areas where access to power and data signaling is limited.

LoRa Radio Technology & The LoRaWAN Network Protocol

LoRa:

LoRa (Long Range) is a Physical layer implementation of a radio modulation method that is based on Chirp Spread Spectrum technology (CSS). It encodes data using chirp pulses and operates in an unlicensed sub-gigahertz band (915 MHz). The maximum allowed bandwidth is restricted to 500 kHz at most, which limits the data throughput of the system, but enables small amounts of data (such as sensor readings) to be transmitted over much greater ranges, with less power, and at a lower cost than alternative technologies such as Wi-Fi, Bluetooth or cellular networks.

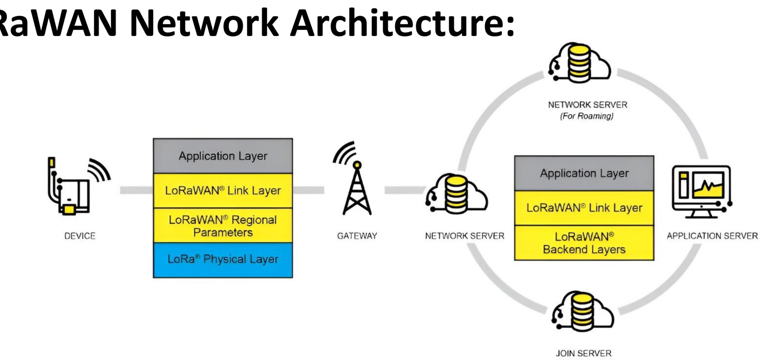
LoRa Compared to Wi-Fi, Bluetooth and Cellular:



LoRaWAN:

LoRaWAN is a Media Access Control (MAC) protocol built for use with the LoRa modulation implementation and is used for creating Low Power, Wide Area (LPWA) networks. The LoRaWAN protocol specifies how devices use the LoRa hardware, including how data is formatted, and how and when data is transmitted and received to and from end devices and gateways.

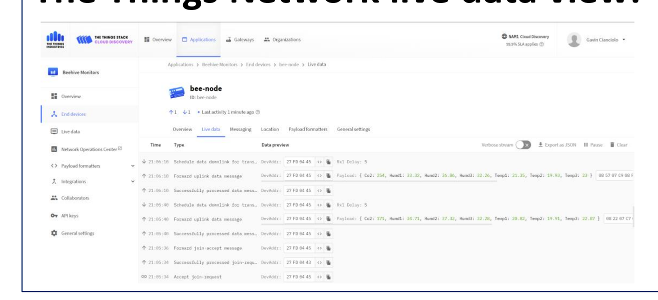
LoRaWAN Network Architecture:



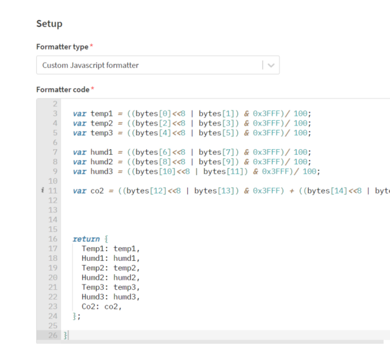
The Things Network

The Things Network is a Software as a Service (SaaS) solution for the Network Server and Application Server components of the LoRaWAN network. Their software and user interface called The Things Stack, bridges the LoRaWAN gateway and whichever end devices the user specifies. The software allows many end devices and gateways to be registered to the same network, handles decrypting network data, and allows data to be parsed, sorted, and sent to different endpoints such as a database, using a variety of methods such as MQTT and Webhooks.

The Things Network live data view:



The Things Network Payload Formatters:



System Overview

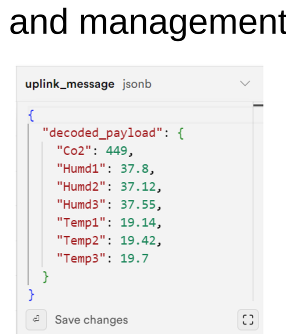
Hardware Overview:

- Adafruit Feather M0 microcontroller:
  - The project is based on the Adafruit Feather M0 microcontroller with integrated LoRa radio submodule.
- Adafruit PCA9548 I2C Multiplexer:
  - Used to allow multiple sensors with the same I2C address to communicate over the same I2C connection
- Adafruit USB / DC / Solar LiPo battery charge controller:
  - Allows LiPo battery to be charged using a USB-C cable or a solar panel
- 5V 5W ETFE Solar Panel
- Sensors:
  - Adafruit HTU21D-F Combined temperature and humidity sensor
    - 5% - 9% relative humidity measurement range, ± 2%
    - -30℃ - 90℃ temperature measurement range, ±1℃
  - MHZ-16 NDIR Co2 sensor
    - 400ppm - 100,000ppm measurement range

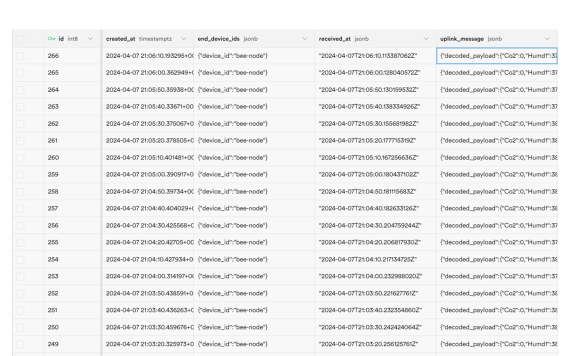
Software Overview:

- Supabase backend server and Postgres database:
  - Takes in data from The Things Application Server via Webhooks and stores sensor data in JSON format.
    - Webhooks are HTTP requests that are used to communicate information between multiple APIs, such as The Things Application API and the Database API
  - Handles database integrations and user authentication
- React Framework:
  - Open source, front-end JavaScript library used to create website / user interface
  - Allows data from sensors to be displayed in a variety of ways and in an easily accessible format
- Vercel:
  - Website hosting and management

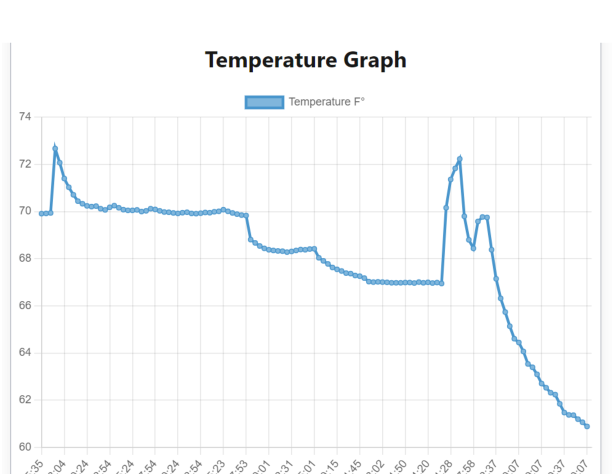
Sensor Data from one packet in JSON format:

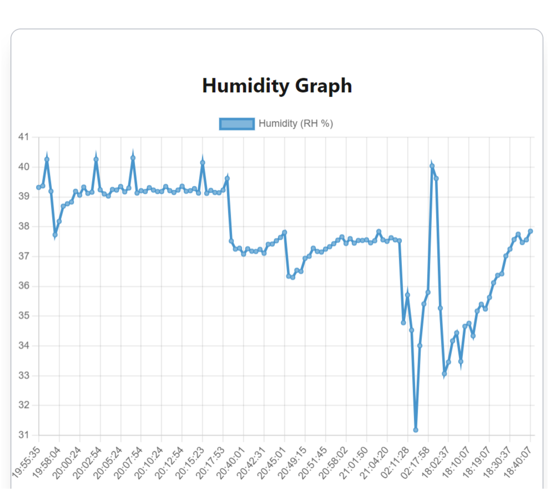


Supabase database table for storing received packet data:



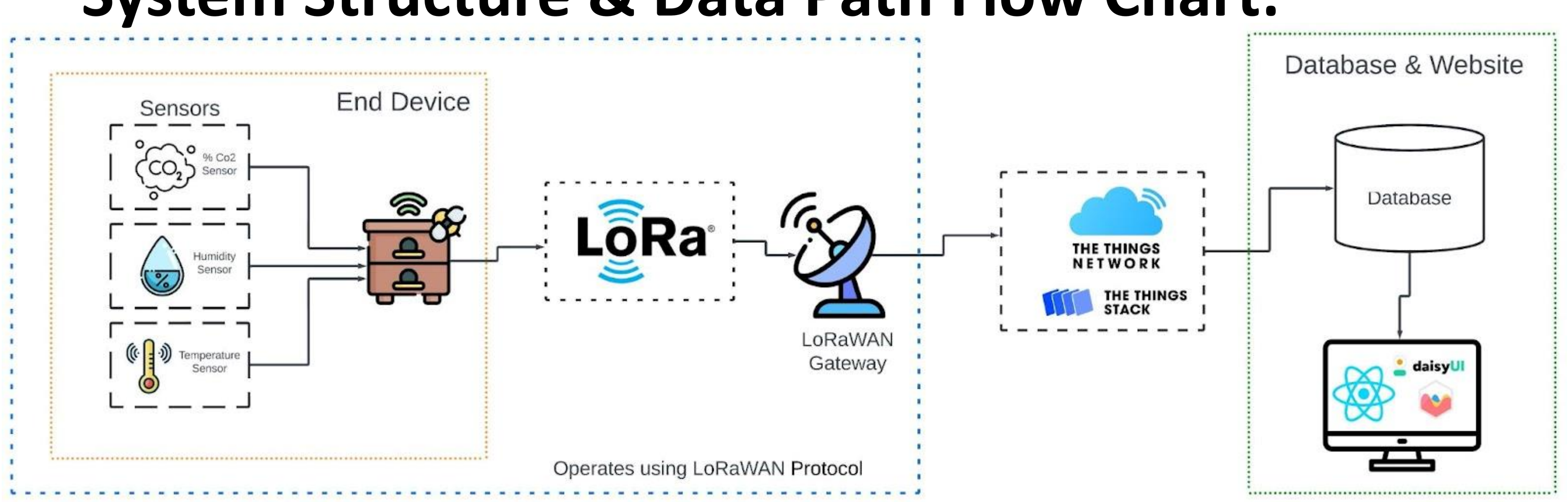
Example temperature and humidity graphs from our website:



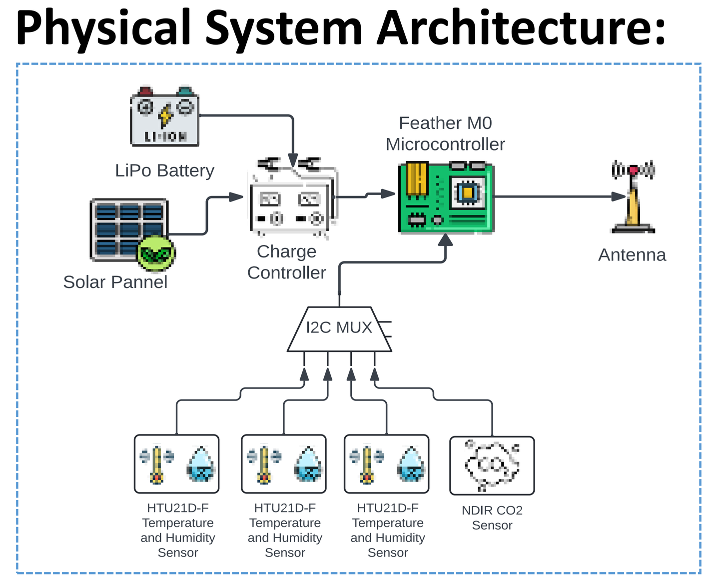


System Architecture

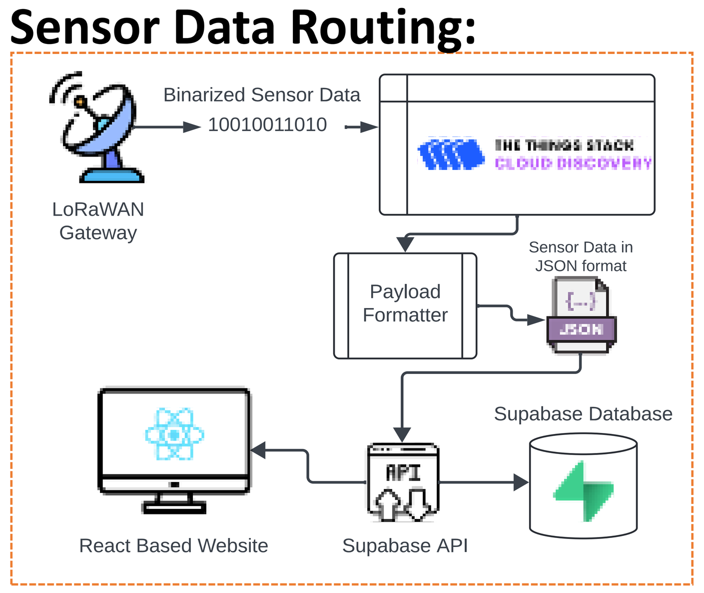
System Structure & Data Path Flow Chart:



Physical System Architecture:

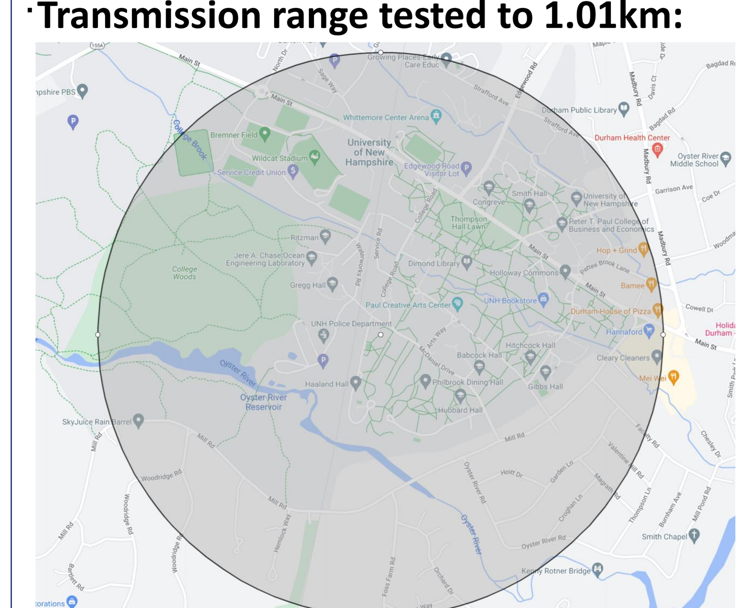


Sensor Data Routing:

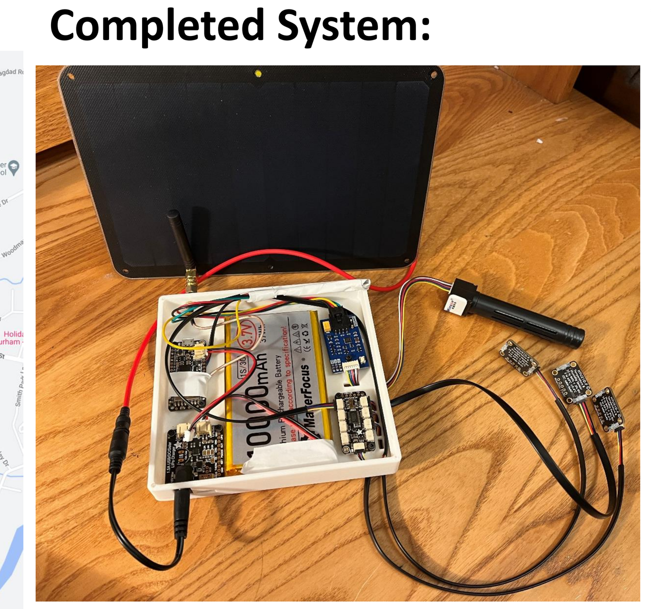


Results

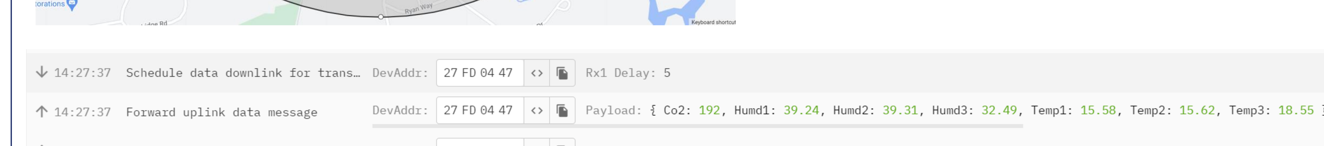
Transmission range tested to 1.01km:



Completed System:



View The Website!



← Successfully transmitted packet at 1.01km from receiver

Next Steps

- Improved Sensing Capabilities:
  - Addition of load cells, positioned in the bottom of the beehive, would allow for weight measurements to be captured, to estimate the number of bees currently in the hive, as well as honey production over a range of time
  - Increasing the number of temperature, humidity, and Co2 sensors in the system would allow for more accurate measurements of climate conditions inside the beehive
  - Addition of microphones to record frequencies generated by the bees
    - Could possibly be used to detect when an invader such as wasps or rodents has entered the hive
    - Could possibly Identify when the bees are preparing to swarm, or when a new queen bee has entered the hive
- Performance & Efficiency Improvements:
  - Implementing a better transmission technique such as Ingenu's RPMA radio technology would allow for greater transmission ranges, lower power consumption, and increased data throughput per end device
- Software Enhancements:
  - Implementing a feature to allow the user to set parameters, which cause the system to alert the user when critical conditions within the hive occur
  - Enabling the user to control the measurement frequency, to allow for more data to be collected, or to conserve power dependent on the individual user's needs
- Durability testing:
  - How durable are the sensors and wires that are placed inside the hive
- Power & Efficiency Testing:
  - How much does the LiPo battery and solar panel degrade over time and what can be done to extend their lifetimes
  - Experimenting with different antennas and antenna placement to maximize broadcast range

Conclusion

We developed a system that remotely measures temperature, humidity, and Co2 levels within a beehive. The system can operate in remote environments utilizing LoRa radio technology to transmit the data over a max tested range of 1.01 km. and can operate in areas that do not have an accessible power source by using a solar panel and LiPo battery, which together powers the entire system. In addition to the hardware component of this project, we have successfully developed a rudimentary database and website for storing sensor data and displaying it in both a text-based and graphical format, which can be extended to allow many users, and many end devices to operate concurrently.

Acknowledgements

We Would like to thank:

Kyle Ouellette & Dean Sullivan for there support during the duration of the project

The UNH IOL for allowing us to use their equipment to complete this project

The New Hampshire Beekeepers Association for the research grant they provided to fund this project

The UNH CEPS Makerspace for their equipment and expertise

Resources:

- lora-alliance.org
- thethingsnetwork.org
- mapdevelopers.com
- supabase.com
- vercel.com
- github.com/lnlp/LMIC-node
- learn.adafruit.com
- nhbeekeepers.org