Team 5

Telemetry, GPS, Solar

Amanda Zhang, Sahana Hariharan, Rebecca Wang

Our team

Amanda Zhang

Junior at Mountain View High School

Sahana Hariharan

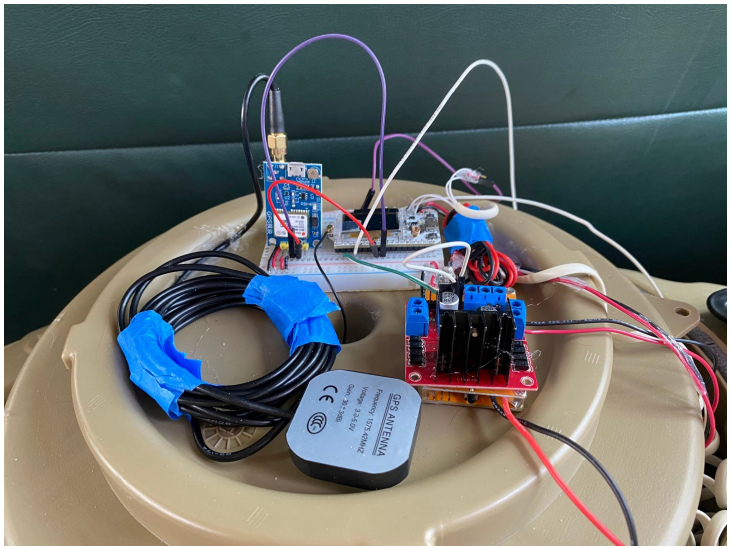
Senior at Notre Dame High School

Rebecca Wang

Junior at Gunderson High School

Abstract

- Collect and send sensor data
- Configure LoRa to work with data acquisition (DAQ) and database teams
- Receive data on server through onshore LoRa and ESP32



Methodology

Sample Code

Learning the hardware.

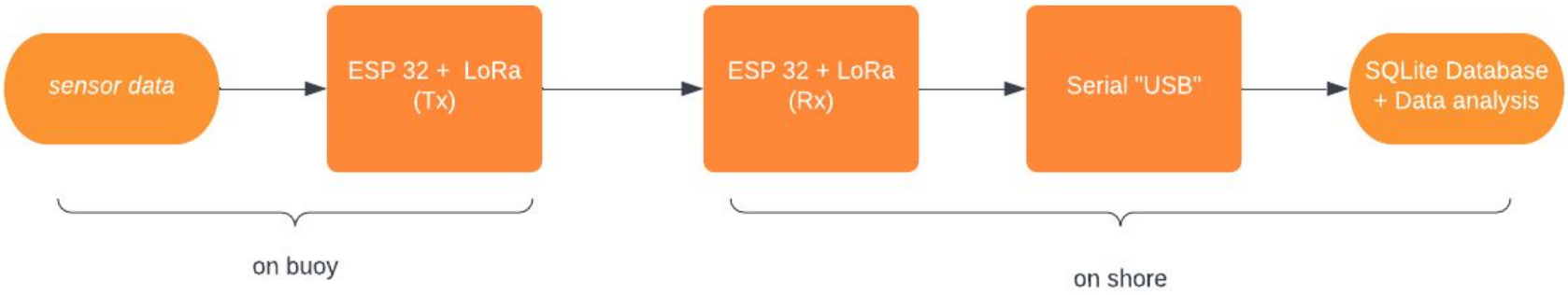
Debugging

Solving software issues.

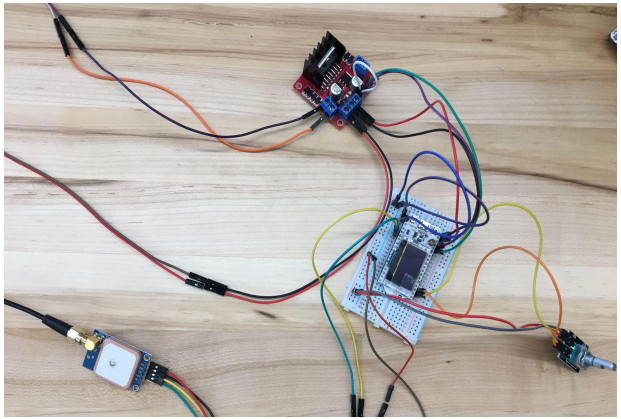
Merging Code

Collaborating with DAQ and Database teams.

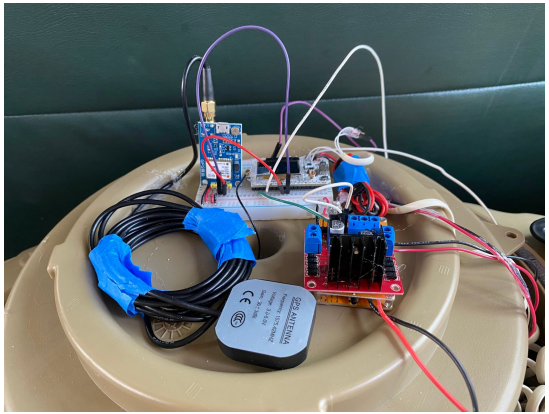
Communication Flow Chart



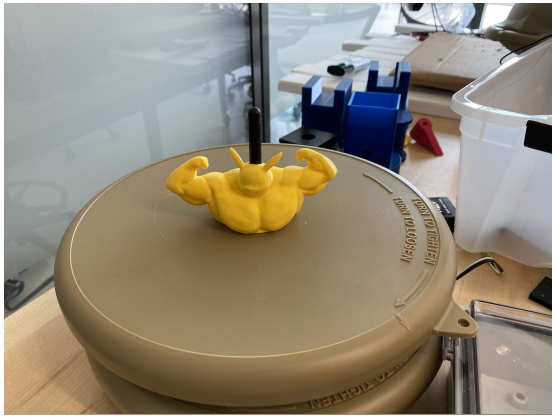
Hardware



Wired GPS, encoder, ESP32.



Final wiring on Buoy 1.



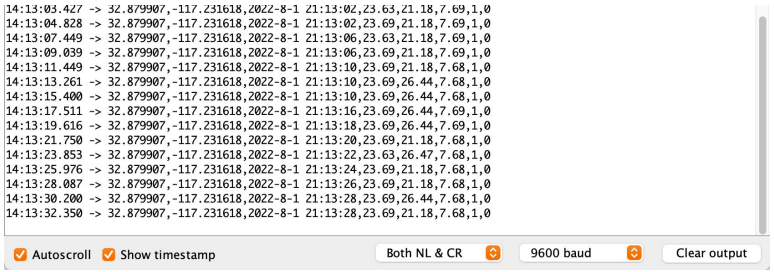
LoRa antenna.

Software

Arduino Libraries: TinyGPS++, SoftwareSerial, and LoRa.

Source Code: github.com/jmtritch/COSMOS22-Cluster13/tree/main/4-telemetry

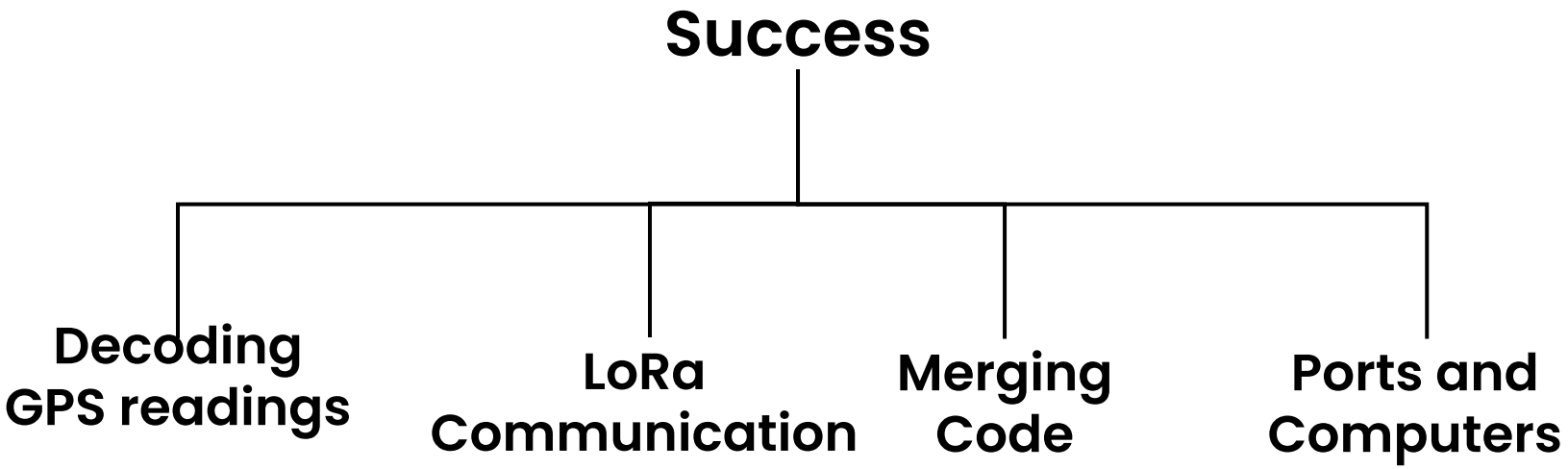
Github QR Code



Serial output.

ESP32.

Challenges


Success
Decoding GPS readings
LoRa Communication
Merging Code
Ports and Computers

Results

Successful

- LoRa radio could send across very long distances
- Miramar Lake - sensor data was successfully transmitted from one buoy to the server

Unsuccessful

- 2nd buoy
- Solar

Future plans

- Test full range of LoRa
- Receive data from capsule buoy
- Receive data from 2 ESP32s simultaneously
- Implement solar

Acknowledgements

Jack Silberman, Ph.D, Ivan Ferrier, J. Michael Tritchler, Dallas Dominguez, Melody Gill, Devanshi Jain

COSMOS

Cluster 13 – H4O: Hacking 4 Oceans

Telemetry

Long Distance Data Communication with ESP32 + LoRa

Sahana Hariharan, Rebecca Wang, Amanda Zhang

COSMOS

Abstract

In the 21st century, remote monitoring of water quality at reservoirs with low power consumption remains a challenge. This project presents a novel automation system receiving and transmitting sensor data from the science buoys to the base station using long range radio (LoRa) technology. We receive temperature, potential of hydrogen (pH), and total dissolved solids (TDS) on our ESP32 microcontroller through serial communication from the Arduino Mega. We added the collected data to our Global Positioning System (GPS) data involving location and time. The combined data, in string format, is sent from the offshore LoRa to the onshore LoRa and then to the server for analysis. Additionally, in the second buoy, the ESP32 controls the depth of the capsule which contains the sensors and Arduino Nano, allowing data to be collected at different depths. Through our work, we communicated the sensors and their collected data from the buoys to the server.

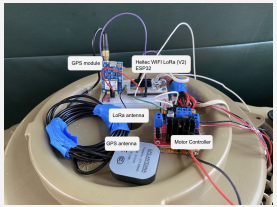


Figure 1. Wired ESP32, LoRa antenna, GPS antenna, motor controller on buoy lid.

Methodology

First, we learned how to use the hardware with Arduino sample code. Then, we calibrated the GPS readings and connected the Data Acquisition (DAQ) team's Arduino through serial to receive their sensor data. Finally, we combined the data into a string and sent it to the server through LoRa and ESP32.

Hardware

We used the Heltec WIFI LoRa (V2) board with an ESP32 to transmit and receive sensor data. Moreover, we used the GPS sensor to collect latitude, longitude, and time.

In Buoy 1, the ESP32 collects location and time data along with temperature, pH, and TDS data from the DAQ team's Arduino Mega. As soon as the sensor data from the Arduino Mega is ready, the DAQ team sends the data as a string to the ESP32 through serial, and the ESP32 combines it with the GPS data. This merged string is sent from the LoRa on the buoy to the onshore LoRa and ESP32. This is subsequently sent to the database for the data analysis team.

Buoy 2 was more complex, as it implemented a capsule that could measure water from depths of 0, 2, 4, and 6 meters, sensor data was serially sent from an Arduino Nano through a telephone wire to the ESP32.

Software

In the Arduino IDE, we used the libraries TinyGPS++, SoftwareSerial, and LoRa. We coded the GPS sensor, while integrating sensor data from the DAQ team into our sender ESP32. Because of the lowering capsule attached to buoy 2, we worked to implement motor controls into our code.

Source Code: github.com/jmtritch/COSMOS22-Cluster13/tree/main/4-telemetry

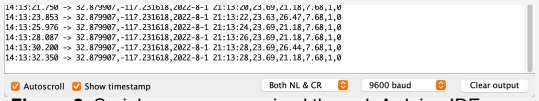


Figure 2. Serial messages received through Arduino IDE.

Challenges

Our team had challenges learning how to use our GPS sensor, especially when decoding the GPS readings. We learned how the baud rate and using GPS within walls affects the GPS outputs. We mainly struggled communicating between the transmitter and receiver LoRa. Furthermore, we had trouble receiving serial messages containing sensor data from the Arduino Mega, which was caused by wiring problems. We also struggled integrating motor controls for the capsule attached to Buoy 2 as it was a new part we had little experience on the electrical aspects.

Results

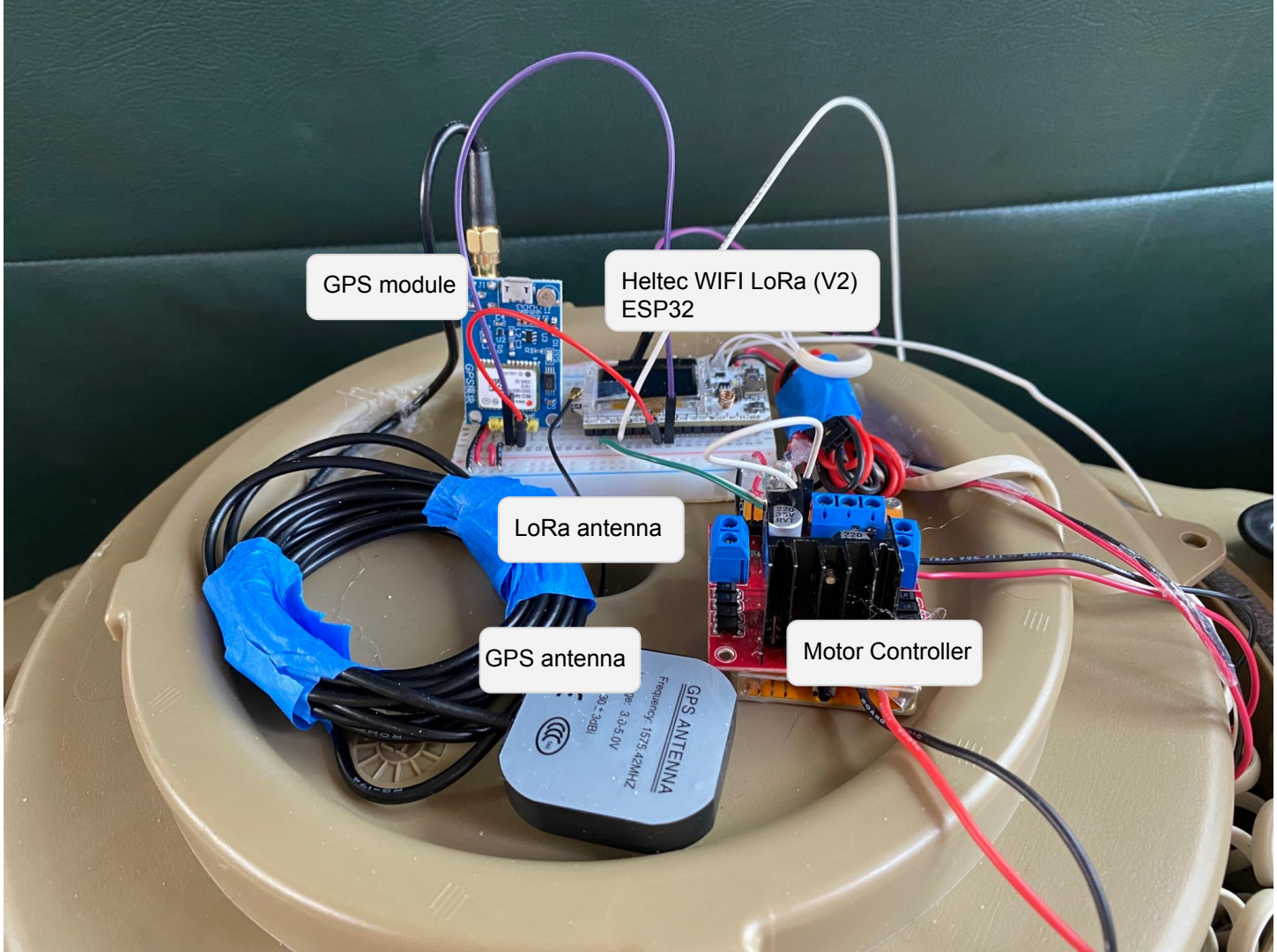
The LoRa radio was able to transmit data across buildings 300 meters apart over the UC San Diego (UCSD) campus. After testing at the UCSD pool and across long distances, we tested our final project at Miramar Reservoir. Sensor data was successfully transmitted from Buoy 1 to the server, even through long distances, natural barriers, and the buoy being half submerged in water. Unfortunately, the second buoy was deficient in power, and the capsule was unable to be pulled up, so we could not test data communication of Buoy 2.

Figure 3. Testing Buoy 2 at Miramar Reservoir.

Acknowledgements

Jack Silberman, Ph.D., J. Michael Tritchler, Ivan Ferrier, Dallas Dominguez, Melody Gill, Devanshi Jain.

Cluster 13 – H4O: Hacking 4 Oceans


GPS module
Heltec WIFI LoRa (V2)
ESP32
LoRa antenna
GPS antenna
Motor Controller
GPS ANTENNA
Frequency: 1575.42MHz
3.0-5.0V
-30~3dBi