Christopher Knauth (email: cknauth119@gmail.com)
Waltrip High School
Instructor: Stephanie Witherspoon (email: swithers@houstonisd.org)

RockeTrack: A compact, long-range, and heavily configurable GPS tracking system

Summary:

RockeTrack marries two existing technologies, LoRa radio and GPS tracking, to create a fully independent live GPS tracking system that operates independently of outside networks and leverages the power of LoRa packet radio to achieve unobstructed ranges of over 5 miles. The system uses two Raspberry Pis, one of which records its location and transmits that data to another, which acts as a receiver.

Hardware:

Fundamentally, the system consists of two Raspberry Pi 3s, each equipped with a GPS transceiver and an RF95 LoRa module configured for 915MHz operation. One Pi is designated the "transmitter" and the other the "receiver" (note that due to the packet radio properties of LoRa, more transmitters or receivers can easily be added to the system).

In the use case the system was created for, the transmitter was mounted in a rocket in a 3D printed mount and powered by a small USB powerbank. Unfortunately, the rocket launch was aborted, and quantitative assessments of the system's performance remain to be performed.

Software:

The system uses Python to interface with the various components of the GPS and LoRa systems, relying on a prewritten C library for sending data. This is done for two reasons: firstly, Python's simpler data types and more user-friendly syntax in general ease debugging and experimentation, and secondly, this modular approach makes using various LoRa transmitters (which might have entirely different commands and protocols) a "plug-and-play" approach rather than requiring a rewrite for each hardware module change. It is true that a singular program to check and transmit GPS data would run with less performance overhead and perhaps with greater reliability, but time constraints made this unfeasible for the project (which was intended to showcase the capabilities of such a system rather than to provide usable libraries for others to implement).

The first component of the software ecosystem is called "gpsd", a native Linux utility which handles RS-232 Serial connection with the GPS and hosts this data on a local server. It also handles conversion from the NMEA data type- the high data rate of which would overwhelm the low-bandwidth LoRa connection- to the much more manageable "gpsd sentences", which contain only necessary information such as latitude, longitude, speed, and heading. Next, the "gpspipe" utility connects to gpsd and writes the data to a file stored locally. This provides an archive of data in the case of the failure of transmission. Finally, the manager python script reads the file, sending any new data to the LoRa sender library, which handles sending.