Balloon Tracking System

Project Proposal: ECE 103

Group Members:

Abdulrahman Alamar [aalamar@pdx.edu]

Ebtehal Alenzi [ebte@pdx.edu]

Chuck Faber [cfaber@pdx.edu]

Project Description

Primary Project

Our primary project will be the payload tracking system for our high altitude balloon, which will include an Arduino as the microcontroller, a radio transceiver (either a DRA818V or a Radiometrix HX-1), and a GPS module which will communicate with the global positioning satellite system. Our payload will also include several sensors that allows us to take readings in the upper atmosphere.

The GPS module will transmit the location of the payload via the radio transceiver to the Automatic Packet Reporting System (APRS) which collects data packets from the US APRS frequency (144.390 MHz) and stores them on the APRS internet server (APRS-IS) which can be accessed by anyone in the world.

Extra Credit

Apart from our primary project, we have several items which we may complete for extra credit should time and resources allow. For example, we may attempt to create an on-the-ground interface system which takes data from the APRS-IS, parses it, and uses it to calculate the direction and distance needed to travel to get to the payload in real time. This "chaser" ground station would be useful once we get to a point where we have to track our payload off the roads.

We may also try to expand upon the sensors included in our payload to bolter our science mission. At this particular time, we are still searching for a good science mission that will benefit actual scientific studies through citizen science organizations. Some possible projects might be helping PSAS test their DxWifi system, or doing atmospheric spectroscopy, or simply atmospheric, temperature, or acceleration sensors.

Some other tasks which might be considered extra credit might be creating a printed circuit board for our balloon payload using EagleCAD, actually flying and tracking our balloon (we intend to do this before the project if we have time, but we should be able to demonstrate the tracking capabilities of our payload even if we do not fly this quarter), and collecting videos and photos of near-space.

Initial Design Approach

Design Goals

Our project will result in a tracking system for a High Altitude balloon, as well as sensors to collect data in the upper atmosphere. Our design requires a GPS module, a radio transmitter, an arduino to serve as the microcontroller, as well as the sensors and associated hardware to collect data. If we have time we will create a "ground station" which will collect data from the APRS-IS system and parse it for use in a real-time tracker that will give us direction and distance to the payload using trigonometric operations, and/or give us data to use for visualizing the full flight path of the balloon.

Furthermore, using the APRS-IS system rather than using only our own singular ground station will allow us to take advantage of the ground station network and repeaters other than only our singular receiver.

Constraints

The modulating and demodulating of APRS (a form of AFSK frequency modulated signal with a 1200 baud rate) seemed to require in depth knowledge of advanced signal processing so we will be using a portion of the FlexTrack library to modulate our callsign, GPS, and sensor data into APRS. Similarly our ground station will use libraries to demodulate the APRS signal.

The majority of the code needs to written by us in C for our project, which means that we need to find the appropriate libraries to use in C for the portions we cannot write ourselves, and that we need to demonstrate a strong understanding in C programming for us to receive credit. We will have to clearly delineate in both our code and our reports which parts of the code were taken from other sources, and which parts were produced by our team.

Our signal should only be transmitted on the APRS frequency about once per minute to allow other traffic to use this frequency as it is used sometimes by emergency responders and other amateur radio enthusiasts.

Using radio transmission for APRS on the ground may produce poor quality signals because of all of the obstacles in the line of sight from the balloon to various groundstations.

Our payload needs to be as light as possible to ensure our balloon reaches high enough altitudes. This can be an issue when it comes to considering the number of items we can add to our payload and also when considering the power source(s) that will likely needed to be added with them.

Assumptions

We will assume that we are the only ones who will be using this tracking system, and so we only need it to be as user-friendly as we ourselves would like it to be. Further, we will assume that our data will be useful to others, and that our program should save location and sensor data in a format that can be easily read and used by others or other platforms (i.e. a comma-separated value file). If we develop the ground tracker/chaser module, we'll assume we need it to be mobile, since we will be running around with it to find the balloon. We will also assume that it will not provide us driving directions, only straight line distances and directions to the payload, and thus should only be used once we are required to travel by foot to the payload.

Design Approach

Our approach to this project will consist of several stages. The first of which is a research stage which will involve us gathering and reading resources on tracking high altitude balloons. This will require us to learn and understand the theory of payload tracking, as well as looking at code and examples written by others in a variety of languages in order for us to understand the kinds of processes and algorithms necessary to write our own code.

Then we will write our own high-level pseudocode based on what we've learned which will describe the general structure of our code and how it will work.

From here we may write a more specific pseudocode which will clarify the C-specific structures we will need to write our program, and also how it will be broken up into separate functions or header files.

Test circuits will be designed for each electronic component and each component will be tested with individual functions to ensure that they work, and that the function works. We will progressively start to connect the electronics with each other, and iteratively test functions in this manner to see that data gets properly sent or parsed. Finally we will test with all of the code and electronics together to ensure that our prototype build and code functions.

This will be done only will our primary project at first. If we find that we have a working prototype that accomplishes the primary function with enough time left over, we will start adding peripherals and the extra credit items, and complete the same testing process.

Task Assignments

Our task assignments are below. For our project, so that each person get experience working in all other aspects of the project, we will all work together on all aspects. However there will be a “lead” or “primary” for particularly subsections of this project. This person will be the go to person for questions in this area, and will manage the addition of these systems to the project by providing the other team members with tasks in this area, and monitoring the progress of this area.

- Abdulrahman:
  - Logistical Tasks: Meeting Facilitator; Meeting notes;
  - Hardware Tasks: Peripheral Primary (i.e. sensors)
    - data sheet research into peripherals
  - Software Tasks: Peripheral Primary Coding for Payload
  - Report Tasks for Proposal: Initial Design Approach
- Chuck:
  - Logistical Tasks: Trello Manager; Bill of Materials
  - Hardware Tasks: Ground Station Primary
  - Software Tasks: Ground Station Primary Coding (i.e. Chaser/Location Parsing)
  - Report Tasks for Proposal: Project Description/Task Assignments
- Ebtehal:
  - Logistical Tasks: Media Director (frequent video and photo evidence); Gantt Chart/Schedule Manager
  - Hardware Tasks: Payload Primary (constructing electronics)
  - Software Tasks: Payload Primary with Arduino and Tracking Code
  - Report Tasks for Proposal: Gantt Schedule

Development Schedule