

By: Sarkis Mikaelian, Nikolaos Stroubos, Leonel Hernandez, Shailesh Sahu

Overview:

1. Goal:

Report telemetry from a self-contained device to a fixed ground station.

2. Components:

- Ground Station:

Arduino -> Antenna/Receiver (Comms) -> Data processing.

- Air Unit:

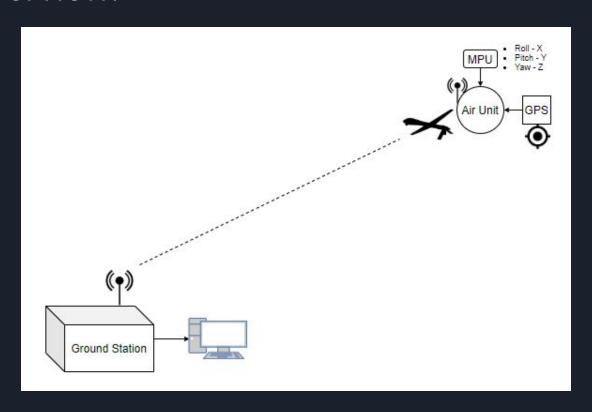
Teensy -> IMU (initial measurement unit) - MPU -> Antenna/Transmitter (Comms).

3. Results:

Data is output to the serial monitor according to the same scheme it's sent to the radio transmitter. Eg.

< A.x, A.y, A.z | G.x, G.y, G.z | M.x, M.y, Mz | AD.x, AD.y, AD.z >

Overview:



Planning:

Software Component -> Teensy & (IMU-MPU, GPS) Code -> Uno & PC Code

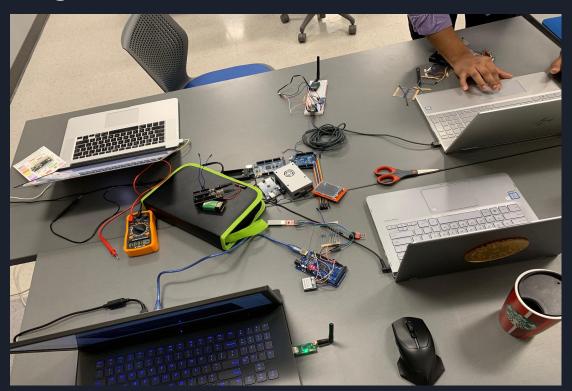
Antenna -> Communication RX/TX Connections and Configurations

Data Processing & Filtering

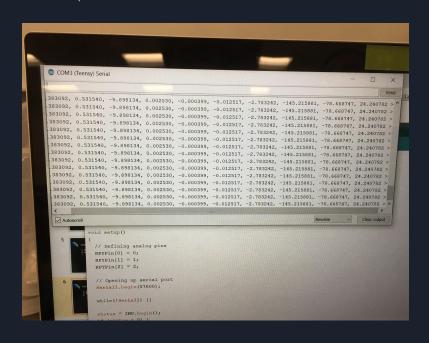
Hardware Component -> Uno - Physical Ground Ops. -> Teensy - Physical Air Unit.

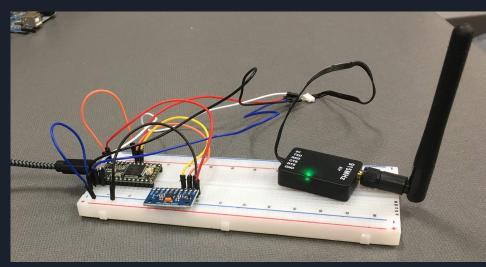
Design & Containers -> 3D CAD SOLIDWORKS

Planning:

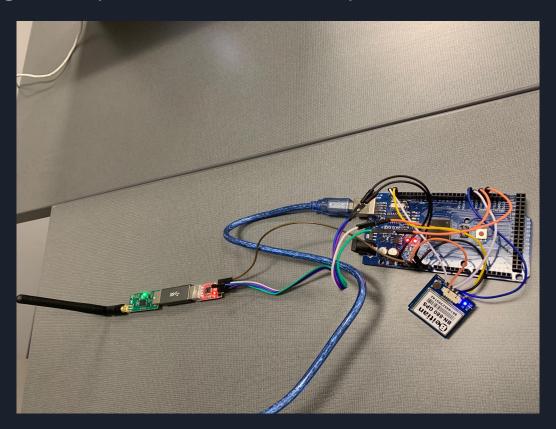


Progress (Air Unit):





Progress (Ground Station):



Data Readings:

```
Hercules SETUP utility by HW-group.com
UDP Setup Serial TCP Client TCP Server UDP Test Mode About
Received/Sent data
The oldest data was removed. Continue...
.000240, 0.000333, -0.000125, 15.481782, -134.569046, -74.466362, 25.142330 >
< 0.426190, 0.521963, -9.878980, 0.000240, 0.000466, 0.000141, 16.351545, -134.394501, -74.298264, 25.133347 >
 < 0.450133, 0.536328, -9.874191, -0.000160, -0.000333, 0.000008, 16.351545, -134.394501, -74.298264, 25.139336 >
 < 0.445344, 0.560272, -9.859825, -0.001491, 0.000200, 0.000407, 16.003639, -135.092651, -73.625877, 25.133347 >
< 0.430978, 0.545906, -9.883768, -0.000426, 0.001132, -0.000525, 17.047356, -132.649124, -73.962074, 25.124359 >
                                                                                                                                                                                                                                  Data size
< 0.435767, 0.521963, -9.874191, -0.000959, -0.000333, -0.001190, 15.829687, -132.823654, -74.466362, 25.127356 >
< 0.435767, 0.526751, -9.855037, -0.000559, 0.000067, -0.000258, 16.351545, -134.045425, -74.634453, 25.121365 >
< 0.407035, 0.541117, -9.898134, -0.000293, 0.001132, -0.000258, 17.221308, -133.870880, -74.466362, 25.130350 >
< 0.411824, 0.565060, -9.878980, -0.001491, -0.001265, -0.000125, 17.047356, -132.998199, -75.306839, 25.124359 >
< 0.440556, 0.531540, -9.864614, -0.000426, -0.000067, 0.000274, 17.047356, -132.998199, -75.306839, 25.124359 >
< 0.435767, 0.502808, -9.883768, 0.000240, 0.000466, 0.000541, 16.873404, -133.870880, -73.793976, 25.136341 >
< 0.450133, 0.536328, -9.864614, -0.000692, 0.000466, -0.000525, 17.221308, -132.125504, -74.802551, 25.130350 >
< 0.416612, 0.536328, -9.883768, -0.000160, -0.000067, 0.000141, 15.829687, -135.267197, -74.130165, 25.115374 >
< 0.435767, 0.517174, -9.888556, 0.000107, 0.000466, 0.000008, 15.829687, -132.474579, -75.474930, 25.127356 >
< 0.407035, 0.526751, -9.931655, 0.002503, 0.000333, 0.000008, 16.351545, -132.649124, -73.962074, 25.124359 >
< 0.430978, 0.531540, -9.917289, 0.001172, -0.000466, -0.000391, 16.177593, -132.823654, -75.474930, 25.115374 >
< 0.407035, 0.536328, -9.907711, 0.000905, -0.000866, -0.000125, 16.177593, -132.823654, -75.474930, 25.124359 >
< 0.421401, 0.521963, -9.859825, -0.002157, 0.000067, -0.000525, 16.525497, -133.521805, -74.802551, 25.130350 >
                                                                                                                                                                                                                                      X Close
< 0.421401, 0.526751, -9.931655, 0.000240, 0.000067, 0.000541, 17.221308, -133.172729, -74.466362, 25.118370 >
< 0.426190, 0.541117, -9.893346, 0.000240, -0.000067, -0.000525, 16.699451, -133.347275, -75.306839, 25.121365 >
< 0.430978, 0.541117, -9.898134, 0.000240, 0.000732, 0.000008, 17.047356, -133.347275, -75.643028, 25.121365 >
< 0.426190, 0.541117, -9.902923, -0.000160, 0.000466, -0.000525, 16.525497, -134.569046, -75.474930, 25.121365 >
< 0.411824, 0.536328, -9.883768, -0.000559, -0.000067, -0.000125, 17.569214, -133.172729, -74.466362, 25.133347 >
< 0.421401, 0.545906, -9.883768, -0.001092, -0.001798, 0.000141, 16.699451, -133.347275, -76.315414, 25.097403 >
< 0.435767, 0.536328, -9.855037, -0.000426, 0.000067, -0.000125, 16.699451, -133.347275, -76.315414, 25.112379 >
< 0.411824, 0.541117, -9.874191, 0.000107, 0.000999, 0.000008, 16.873404, -132.125504, -74.802551, 25.124359 >
< 0.421401, 0.541117, -9.893346, 0.000506, -0.000866, -0.001324, 15.829687, -132.474579, -74.130165, 25.115374 >
 0.407035, 0.536328, -9.907711, 0.000905, -0.000866, -0.000125, 16.177593, -132.823654, -75.474930, 25.124359 >
< 0.421401, 0.521963, -9.859825, -0.002157, 0.000067, -0.000525, 16.525497, -133.521805, -74.802551, 25.130350 >
< 0.421401, 0.526751, -9.931655, 0.000240, 0.000067, 0.000541, 17.221308, -133.172729, -74.466362, 25.118370 >
< 0.426190, 0.541117, -9.893346, 0.000240, -0.000067, -0.000525, 16.699451, -133.347275, -75.306839, 25.121365 >
< 0.430978, 0.541117, -9.898134, 0.000240, 0.000732, 0.000008, 17.047356, -133.347275, -75.643028, 25.121365 >
< 0.426190, 0.541117, -9.902923, -0.000160, 0.000466, -0.000525, 16.525497, -134.569046, -75.474930, 25.121365 >
< 0.411824, 0.536328, -9.883768, -0.000559, -0.000067, -0.000125, 17.569214, -133.172729, -74.466362, 25.133347 >
< 0.421401, 0.545906, -9.883768, -0.001092, -0.001798, 0.000141, 16.699451, -133.347275, -76.315414, 25.097403 >
< 0.435767, 0.536328, -9.855037, -0.000426, 0.000067, -0.000125, 16.699451, -133.347275, -76.315414, 25.112379 >
< 0.411824, 0.541117, -9.874191, 0.000107, 0.000999, 0.000008, 16.873404, -132.125504, -74.802551, 25.124359 >
< 0.421401, 0.541117, -9.893346, 0.000506, -0.000866, -0.001324, 15.829687, -132.474579, -74.130165, 25.115374 >
```

Questions:

- How should each axis of data be mapped?
- What communication protocol is each component of the project using?
- How will the data be collected from the Arduino?
- How does the Beitan GPS work? How do we print data from it to the serial monitor repeatedly?
- How does the Serial*.available() function work?
- How many devices can the Mega handle?
- At what baud rate is the data optimally received at from each device?
- What have we learned so far?
- If we were to receive data from the computer, how would it be graphed?

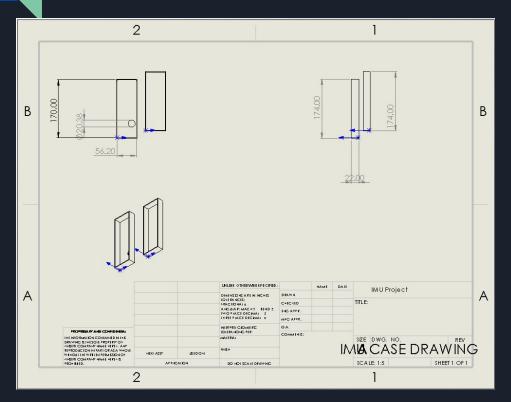
Challenges:

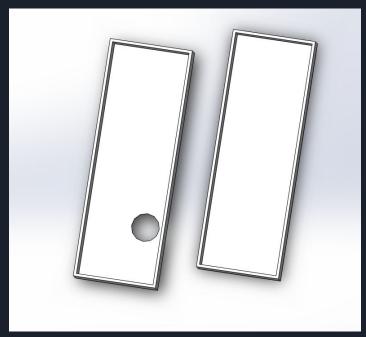
- Team Coordination.
 - Worked on better communication.
- Arduino Malfunction (UNO R3).
 - Replaced it with a Mega 2560.
- USB to TTL Problems (USB Slave).
 - Attempted to replace the radio telemetry unit with 3DR Radio
 915MHZ with serial connections on both ends.
- Dealing with Arduino Libraries when reading from both MPU & GPS.
 - Need to merge the two code bases and deploy on Mega.

Future Work:

- 1. Building a container for both Air and Ground Units using Solidworks.
- 2. Setting up the Air Unit on a drone and run tests.
- 3. Connect multiple sensors, potentially a camera.
- 4. Continue answering the questions on slide 9.

IMU Case-Ground





References & Reproducibility:

GitHub:

https://github.com/SMikaelian/BATTS

Book Reference:

https://learning.oreilly.com/library/view/arduino-cookbook-3rd/9781491903513/ch01.html#getting_started