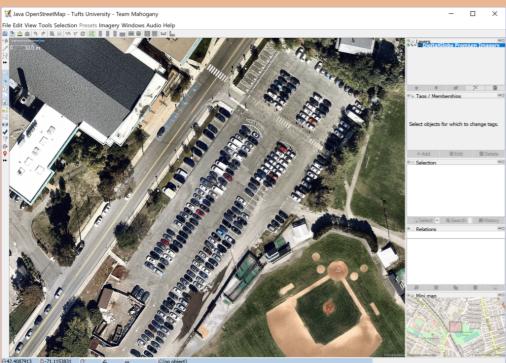


User Interface

Since most open source libraries are built for post processing images, we struggled to enable them to work in real time. In addition, overlaying collected data on top of an existing map is challenging because the system must work in an offline setting.



Processing Received Data

After receiving the 60 byte samples, the user interface converts the samples back into a JPG image. Below is an example of a Hex editor, used to verify that received images were correct, and convert back into JPG form.

UAV Search and Rescue

Data Collection

We created a black box system to provide first responders with real time data about a site after a disaster. The first step was to collect data. Using the tools pictured below, data was collected in packets, and saved externally to a microSD card on board the system.

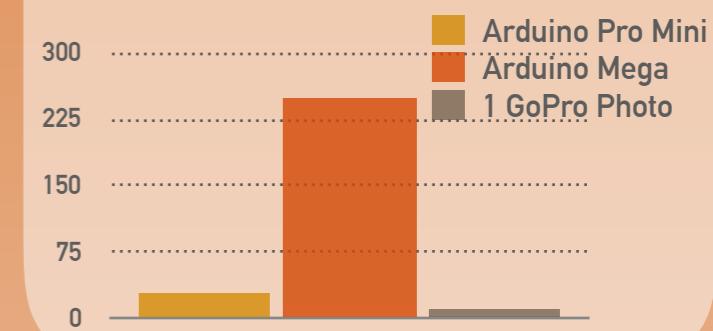


- a. GoPro Hero 3
- b. GPS Breakout
v3 by adafruit
- c. Geiger-Muller
Tube

Data Storage

Each image was saved as a packet to a MicroSD card and configured into payloads. We had limited storage due to the small arduino we used.

Arduino Storage compared with GoPro image (in Kilobytes)



Real Time Transmission

Real time transmission was the largest challenge in our project. By splitting up each image into 60 byte samples, each sample could be sent and received by the receiver module.

