University of Minnesota – Twin Cities Modifications to the Montana State University Telemetry System for Stratospheric Eclipse Ballooning

Benjamin S. Geadelmann[1], Austin J. Langford[2], Austin N. Eiler[3], Ryan Bowers[4]
University of Minnesota-Twin Cities, Minneapolis, Minnesota, 55455, USA

James Flaten[5]
MN Space Grant Consortium (MnSGC) and U of MN – Twin Cities, Minneapolis, MN, 55455, USA

On August 21, 2017, the path of totality of a solar eclipse swept across the continental United States from Oregon to South Carolina. Our team, flying weather balloons near Grand Island, Nebraska, was able to live stream the shadow of the moon from the stratosphere to the ground. The team was able to track our balloons with high accuracy due to new payload software and hardware implemented on the still image telemetry platform developed by the Montana Space Grant. In addition, the modified system allowed the team to relay commands and receive information from individual payloads attached to our balloons, giving live telemetry and control from a new GUI-based ground station control application. Although the eclipse is now over, the system will still be a powerful and useful tool for the University of Minnesota stratospheric ballooning team. The platform could be used for any other application needing real-time, ground-based communication to various payloads on a balloon gondola.

[1]Undergraduate student, Aerospace Engineering and Mechanics, geade004@umn.edu.
[2] Undergraduate student, Aerospace Engineering and Mechanics, langf063@umn.edu.
[3] Undergraduate student, Aerospace Engineering and Mechanics, eiler084@umn.edu.
[4] Undergraduate student, Aerospace Engineering and Mechanics, bower290@umn.edu.
[5] Associate Director of NASA's MnSGC and Contract Associate Professor, Aerospace Engineering and Mechanics, flate001@umn.edu.

I. Introduction

THIS past August over 50 teams from across the country worked together to attempt to live stream the total solar eclipse that occurred from coast to coast. Montana State University (MSU) led this effort and developed a Raspberry Pi-based system for live streaming video from balloons using 5.8 GHz radio telemetry. In January of 2016, representatives of the University of Minnesota – Twin Cities' (UMTC) stratospheric ballooning team travelled to Bozeman, Montana, to learn more about their video-telemetry system and to become a "leadership" team for this national Space Grant project.

After building MSU's system and testing it on balloon flights in Minnesota, our team believed that some changes should be made to the system to allow the best chances of success during eclipse missions. Thus began the "Minnesota tracking system" which was built to improve the GPS update time of the MSU tracking system, to allow for down range tracking operation of ground stations, to relay commands to different payloads, as well as to improve the overall usability and robustness of the system. The Minnesota tracking system added functionality to the still image payload designed by MSU, making it useful for tracking and also serving as a communication relay with up-link and down-link radio capabilities.

The original still image payload contained a Raspberry Pi, RFD900 radio modem, and a Raspberry Pi camera. Our team decided to use the existing 900 MHz radio link between the ground station and this payload as the basis for a new communications relay system. This payload needed to be able to relay commands from the ground station to the Raspberry Pi. The Pi in turn would send these commands to the desired payloads on the balloon stack by short-range XBee radio. Other needs of the communications relay were to know its GPS location at all times and to be able to send messages to the ground station for tracking, payload management, and general status updates, both autonomously and when commanded.

II. Additional Hardware

A. Flight Hardware

The first challenge of this project, after deciding to use the existing RFD900 radio connection, was to select the extra hardware needed. The first component selected was an Adafruit Ultimate GPS Breakout. It was used for tracking of the balloon. This board was compared with other GPS modules to test performance, reliability, and features. Our team tested the Adafruit module, a UBlox NEO6MV2 module, and a Trimble Copernicus II. The

Ublox receiver was cheapest, but it had no internal memory and could not retain the required high-altitude state when it was turned off. The Copernicus was the most expensive module. It was extremely configurable, but we decided against it due to the cost compared to what we wanted it to do. The Adafruit module seemed to be exactly what we wanted. It was not too pricey and relatively easy to use. Unfortunately, the firmware of some newer Adafruit boards will not allow GPS signals to be read over 18,000 meters (about 59,000 feet). The older Adafruit modules did not have this altitude limitation, and worked well. In the end, we primarily used the Adafruit GPS modules when available, though a fix for the UBlox memory issue was also explored and lightly tested.

To connect the GPS to the Raspberry Pi, our team decided that the simplest and most effective way was to pass the GPS serial output through a USB-Serial converter. This allowed the team to easily parse information coming in, along with providing a secure attachment method, as the USB ports have excellent mechanical grip. Additionally, using USB-Serial converters made it easy to determine if a device was connected by checking the OS serial port connections.

For inter-stack communication, an XBee radio was also added to the communication relay. XBees were used because our team already had several of these radios and knew that their capabilities would be appropriate in this project. The XBees were connected to the Pi via USB dongles due to the similar advantages this attachment method offers to the USB-serial converters. See Figure 1 for a picture of an early variant of the still camera payload with a GPS breadout board and an XBee radio added.

Other changes made to the payload included changing the connection of the RFD900 from GPIO pins to USB, and powering the Raspberry Pi through the micro USB port to prevent damaging the board. In order to use the new communication relay with our other payloads, we redesigned the MSU code with a multithreaded approach to process information travelling in both directions through the RFD900 and XBee network, from payloads to the ground and from the ground to payloads, allowing the communications relay to combine both radios. Arduinos, Teensies, and other microcontrollers on the stack were now reachable by radio from miles away using only their low-powered XBee radio modules. This conversion was done on many of the team’s payloads including vent-arrow (to achieve float), a multicut box (to sever multiple lines on command), CHAD (for automatic heading control), atmospheric characterization (with temperature, pressure, and dust sensors), and GPS-comparison boxes (use when testing the Adafruit modules against the other types of GPS units mentioned above). These functionality upgrades now allow the ballooning team to collect and examine sensor data in real time, to activate mechanical components using ground commands, and to check system status during flight.

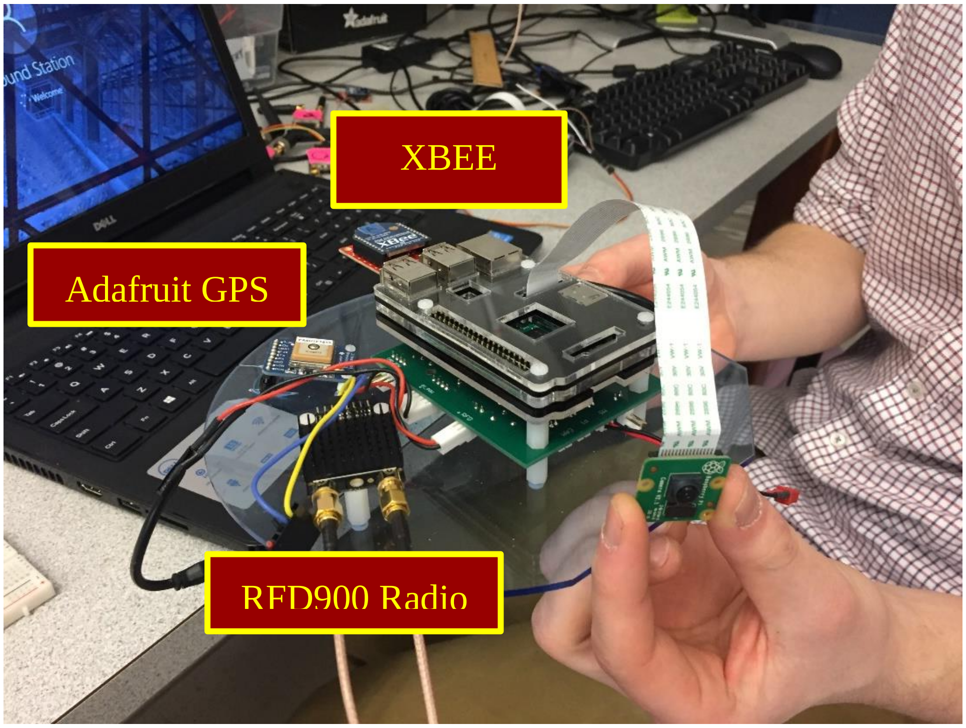

Fig. 1 An early version of "Communications Relay" with GPS breakout and XBee radio.

B. Ground Station Hardware

The ground station hardware originally developed by MSU was left largely intact. The main change to this part of the system was the location of the ground station, complete with tracking antennas, during flights. Our team updated the firmware of the radios and made software changes to make the flight code autostart in the communications relay and video-telemetry payloads. Being able to track balloons from "downrange" (near the predicted burst location) increased the reliability of the telemetry system by making the distance between the balloon and the ground station shorter, sometimes much shorter, during the peak of the flight than had we left the ground station "up-range" (i.e. near the launch site).

Downrange ground stations, though they do require additional people to staff them who are not at the launch site so cannot assist there, also allow more flexibility for location scouting. The ground station needs a reliable, high-speed internet connection in order to stream video. It also needs power and a clear view of the sky. These requirements, coupled with the requirements for a good balloon launch location, often don't occur all in the same physical location.

Setting up ground stations downrange also solved a problem of the balloons drifting out of range of the ground station antennas during crucial/high-altitude parts of the flight The only additional piece of hardware needed for the our downrange stations is an omnidirectional 900MHz, 6 dBi antenna from RFDesign, the linear white antenna shown in Figure 2. The omnidirectional antenna allowed our team to pick up the balloon's signal without even having to point the ground station antennas at it. The omnidirectional antenna was usually able to connect to the balloon's RFD900 when the balloon was above 20,000 feet even if the balloon stack was up to 60 kilometers away.

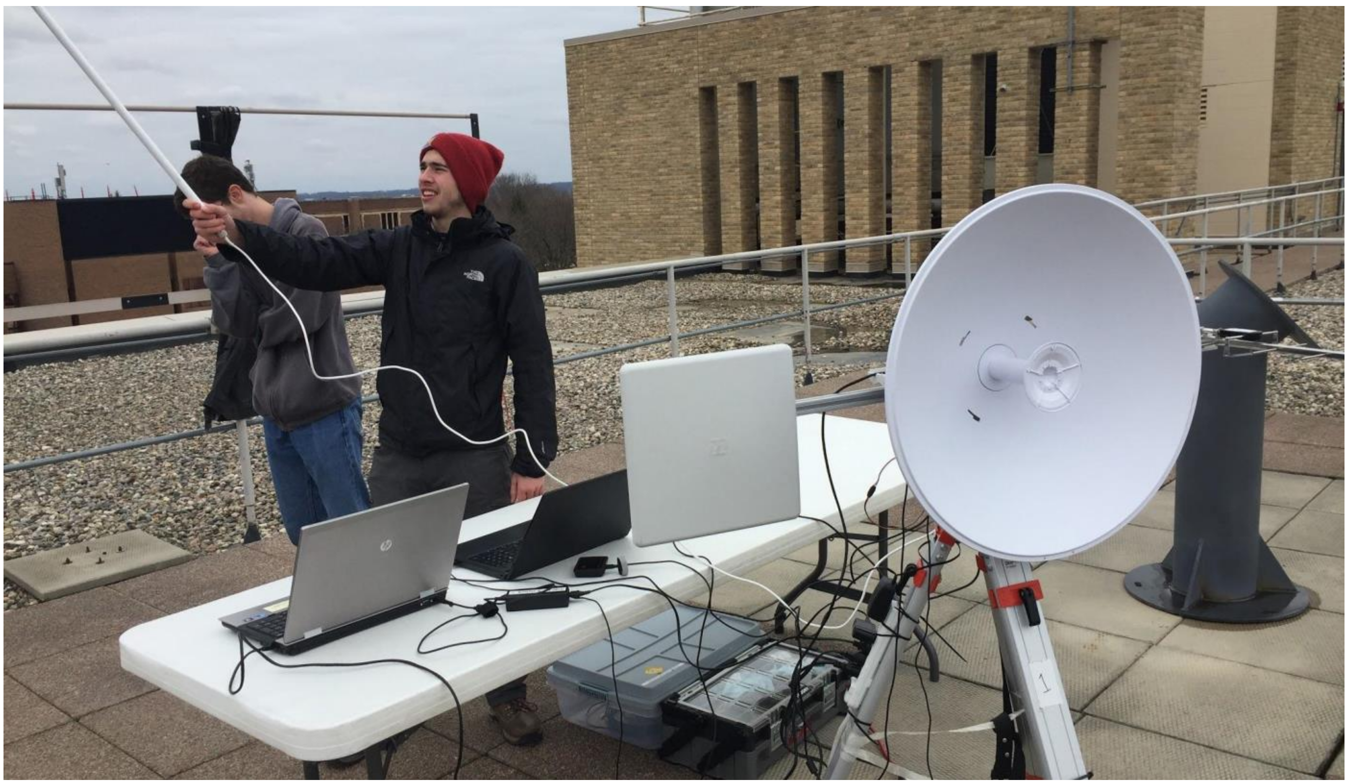

Fig. 2 UMN team member operates the downrange ground station with omnidirectional antenna.

III. New Software

The Minnesota modifications to the software packages initially developed by the Montana Space Grant began in earnest during the summer of 2016. The team decided that we wanted to expand the capability of the system to become an overall balloon communication system, with both uplink and downlink capabilities. To accomplish this task, the still image reception software was ported into the ground station control software to create a single unified ground station control and radio control program (see Fig. 3). At the same time, modifications were made to the still image system flight software to enable the relaying of information between local payloads and the ground station.

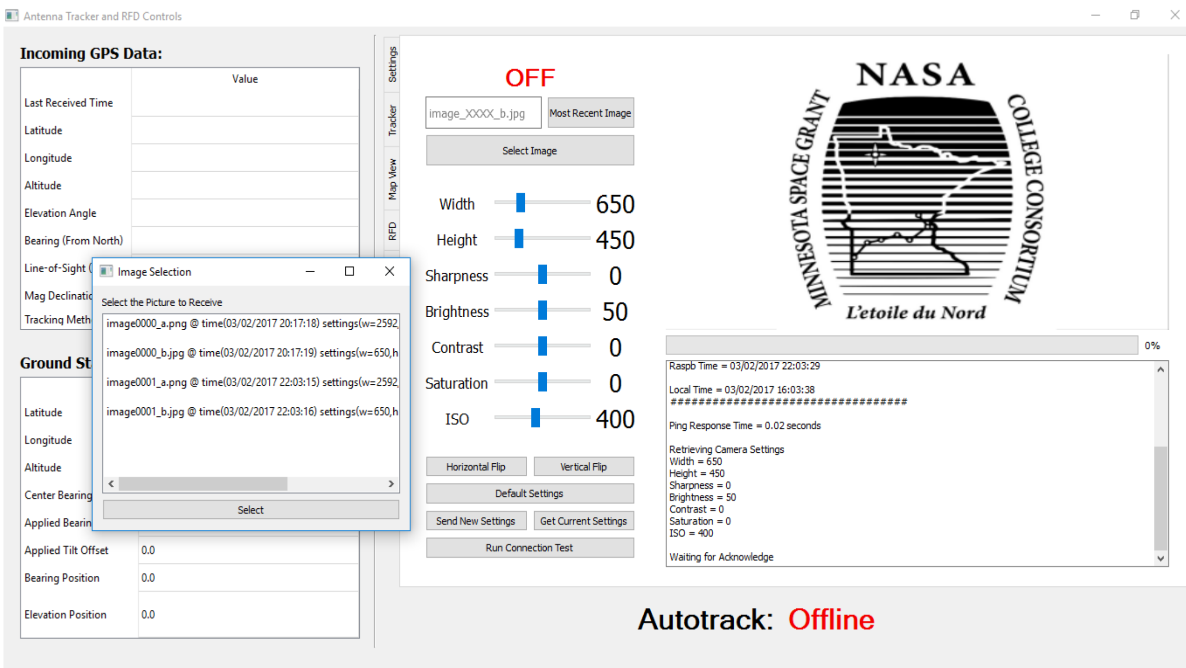

Fig. 3 The Still Image System ported into the unified ground station control software.

A. Ground Station Software

The ground station was expanded to accommodate multiple tracking inputs from a single balloon simultaneously. We found that GPS updates every 30 seconds through the Iridium tracking system distributed by MSU wasn't adequate to maintain a reliable video link to the ground under all flight trajectories. Currently-supported methods are Iridium, Direct (via RFD 900 broadcasts), APRS, and GPS packets coming to the communications relay by XBee message from other payloads on the gondola. The ground station software is now multithreaded, to better-support these activities while leaving the interface responsive at all times.

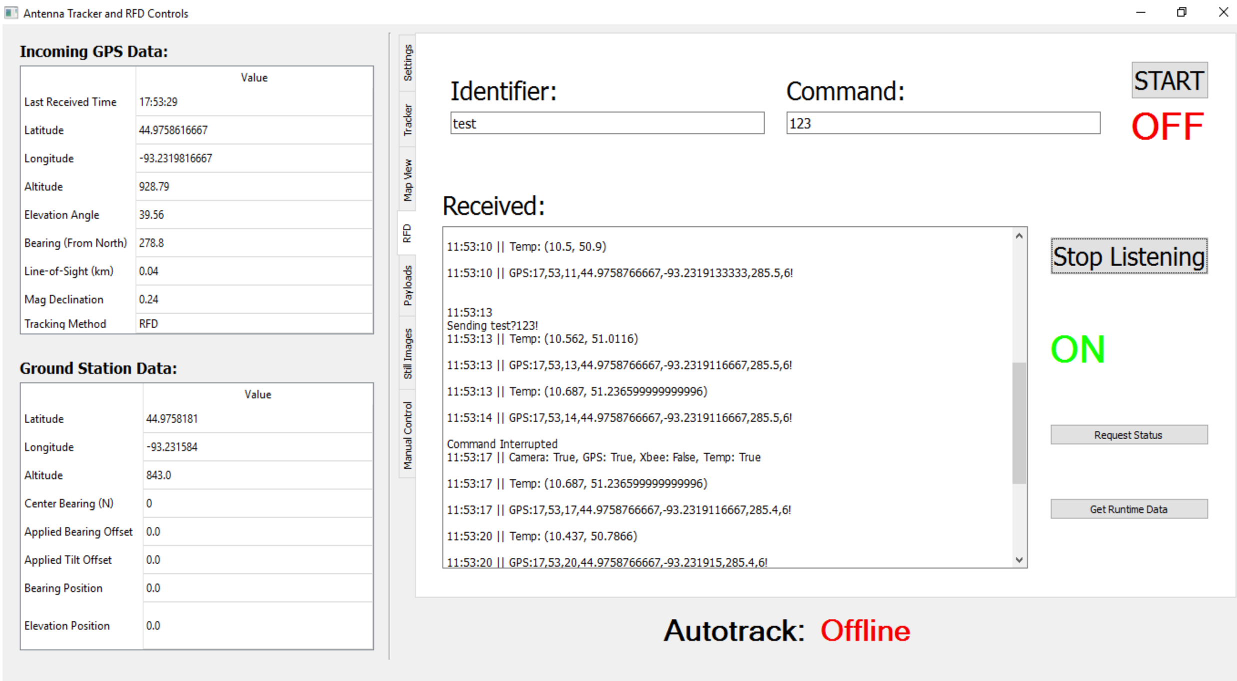

Fig. 4 The RFD tab allows commands to be sent and also displays messages received.

The new RFD tab (see Fig. 4) has an interface to broadcast commands and a window to show all information that has been received. There are additional command options such as (a) a communications relay status request, which returns a list of devices that are connected, (b) a runtime data request, which returns the runtime log of the communications relay itself, and (c) cut-down controls. The payloads tab (see Fig. 5) separates information according to specific payloads, organizes it, and hence allows the ground station staff to monitor communications with each payload independently. The ground station tripod servo mapping was reworked to allow for easier swapping of new servos with different mapping.

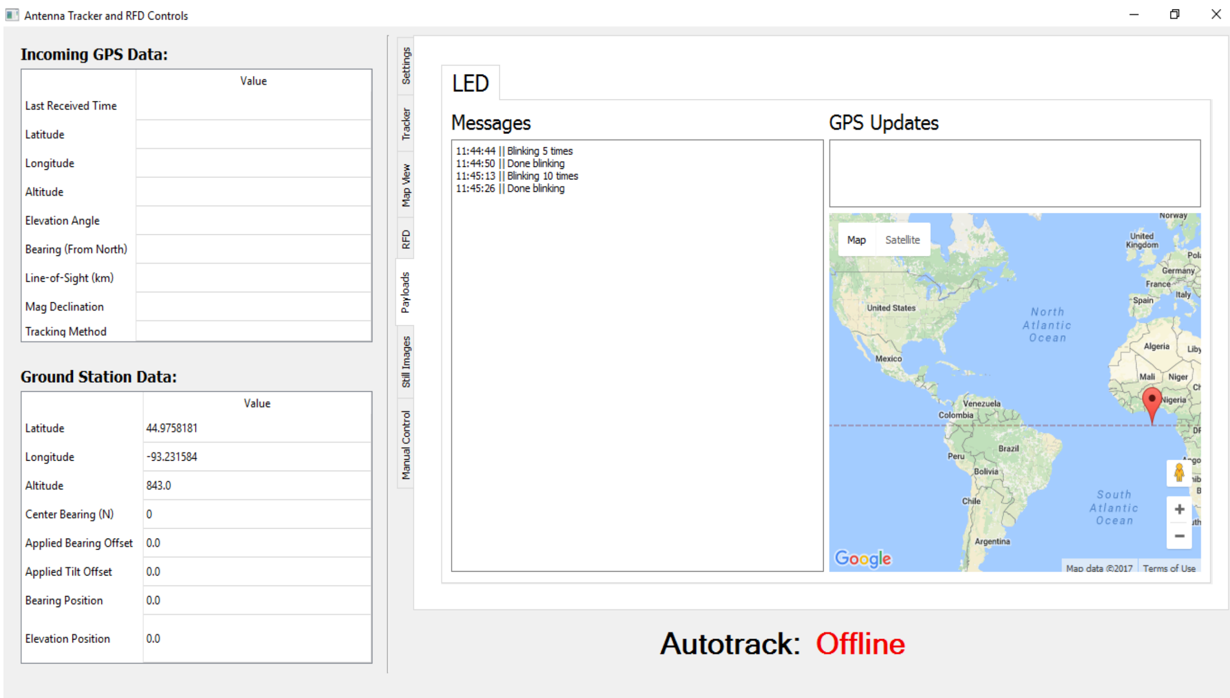

Fig. 5 The Payloads tab organizes information received from each unique payload

The map tab dynamically adds the location of the balloon to a Google Maps plugin as GPS packets are received, creating a map of flight path in real time. Additionally, we made other smaller updates to make the ground station software to make it more easy to use such as automatic serial device detection, a new IMU calibration interface, warning pop-up windows for various issues needing attention, and manual controls for the servos (see Fig. 6).

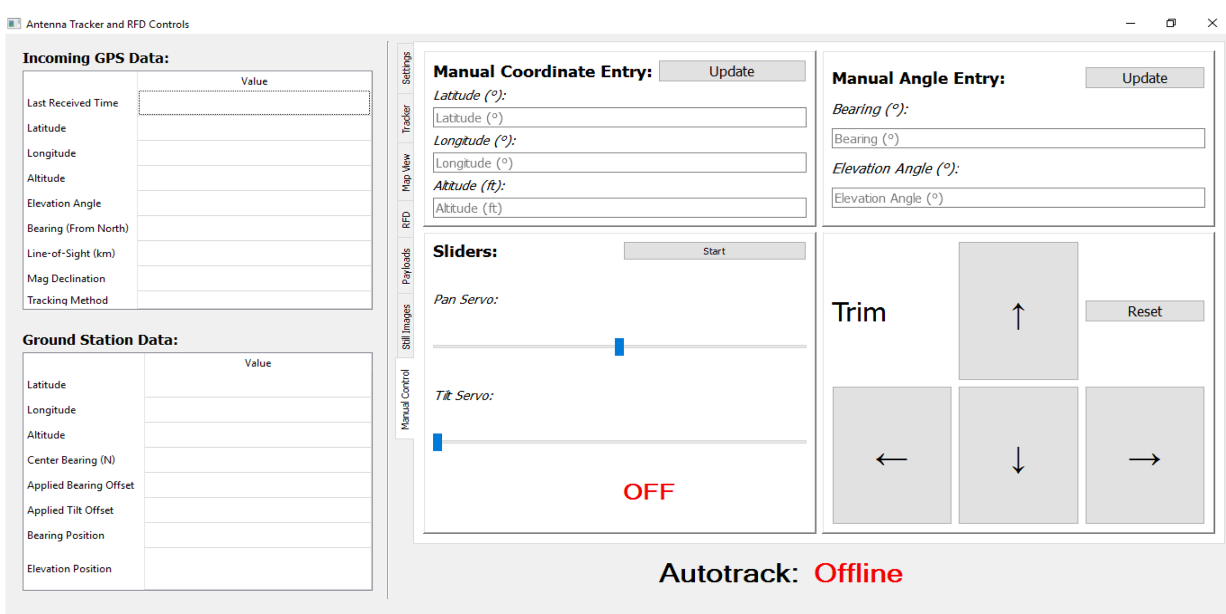

Fig. 6 Manual Servo Controls

B. Flight Software

The still-photo payload flight software was modified to take on a completely different purpose than originally intended. The new primary mission of the flight software is to beacon the payload's GPS location every 3 seconds and to relay information sent via XBee radios in nearby payloads all the way to the ground station. The program is multithreaded to allow for simultaneous handling of radios, as well as continuously retrieving new GPS locations, and also taking still pictures. The software is highly error tolerant, with the ability to remove any external device in flight or to accept additional devices being attached at any time. It is capable of triggering a balloon cutdown by broadcasting the OCCAMS cut signal independently of the Iridium network, or by signaling an altogether different cutdown method. A detailed explanation of this feature can be found in Ref. [2].

III. UMN Findings and Experiences

Over the course of almost 2 years of development and testing, our team has generated countless stories and discovered as many ways to conduct, and some way to not conduct, balloon operations. The team's experience has been a roller coaster of sorts, and we have developed the ability to adapt to whatever a flight may bring. We have developed an understanding of how the student-designed-and-built equipment works, and all of the quirks and subtleties that they have over the course of approximately twenty test flights, most with several balloons. Every week brought its own challenges and the team learned and became more comfortable with the equipment as time went by. For example, by trying different operating procedures we ultimately made the decision to locate our ground stations downrange of the launch site (closer to the predicted burst point), and also to pick up the signal from the balloon at around 20,000 ft, allowing a better chance for maintaining a live video stream to the peak of the flight.

As the eclipse day approached, the team faced its largest dose of adversity. One week before eclipse day, on a full scale test flight in Minnesota, two of our three eclipse telemetry stacks landed in two separate lakes. Both of the stacks were completely soaked and the decision was made to immediately desiccate all electronics in rice, to try to get moisture out of the circuitry. Over the following week, the team was able to replace parts that had been too badly damaged by the water exposure, and to salvage parts that couldn't be replaced to get all three systems back to 100% operational capacity. If it had not been for the amount of time that the team had put in getting comfortable with the equipment, and the level of understanding that the team had developed because of how familiar we were with the equipment, the operations on eclipse day would not have gone as smoothly as they did. This comfort with the flight systems allowed team members to know that we were as prepared as possible for eclipse day (see Fig. 7).

Fig. 7 Prepping the five balloons on eclipse day.

On the day of the eclipse, the team was launched five balloons, three of which were eclipse-telemetry balloons. For each telemetry stack, there was an associated ground station. The launch location was in Kenesaw, Nebraska, and there were ground stations in Grand Island and Aurora. The flight prediction had the balloons flying directly south of both ground stations, and then landing near Aurora. Ground Station A (in Grand Island) had a hard-wire internet connection and two team members operating it. Ground stations B and C (in Aurora) had a "private" wireless internet network to upload video with four team members operating the two stations. The balloons were launched between 11:30 a.m. and 12:05 p.m. from Kenesaw. When stack A was launched, ground station A immediately picked up RFD900 tracking signal and shortly thereafter established a solid video connection. Ground stations B and C picked up RFD tracking from stacks B and C respectively, but the video signal took quite a while for those stations to pick up because Aurora was 20 miles farther away from Kenesaw than Grand Island. Unfortunately, the balloon carrying stack A and being monitored by ground station A (in Grand Island) popped about 15 minutes before totality. This was unfortunate because stack A carried the team's best equipment and ground station A had the best internet connectivity. As totality approached, both stacks B and C were getting video to the ground and up to the internet. However about 10 minutes before totality the internet connection was lost on both stacks B and C, likely due to the private wireless network getting overwhelmed. After the stacks returned to the ground, it was found that the on-board video recovered was good, just what we were trying to stream.

IV. Future Work

Although the eclipse is now past, our modified telemetry system will remain very useful. This system gives our team the ability to communicate with balloons in flight in a way that we never had before. Now we can receive data during flights and command mechanical components and other tasks at any time, including at altitude. This gives the team more flexibility for building payloads as well as adds redundancy to our missions. For example, in a new project to float a balloon at a certain altitude, we can directly command the flight-termination cutter instead of relying on timers and autonomous actions. We can also get feedback from the system to let us know if cutting attempts were successful. Even if no active commanding is required, this communications system lets our team get status updates from payloads to ensure they are working properly. Another future goal is to make this system even more flexible and easy to use. If no video telemetry is required for a flight, our team has been able to establish and maintain an RFD900 telemetry link with only a computer, a ground RFD900, and an omnidirectional antenna. If this “minimal” ground station is in a vehicle driven below the balloon’s flight path while the balloon is in flight, it is usually possible to stay close enough to the balloon to maintain radio connection for the majority of the flight. One possible improvement to this system would be to use a car-mounted antenna for the mobile ground station. This would likely hold the telemetry connection even longer than the present method of keeping the omnidirectional antenna inside of the car or holding it outside the car window by hand.

References

[1] Flaten, J., et al, “Eclipse-Ballooning 2017: The U of MN – Twin Cities Experience” *oral presentation and paper in conference proceedings*, AHAC 2017, Minneapolis, MN, 2017

[2] Ailts, G., Peterson, S., Toth, D., Nelson, J., and Flaten, J., “Rotation Mitigation and OCCAMS/Tungsten Flight Termination for Eclipse Balloon Missions,” *poster presentation*, AHAC 2017, Minneapolis, MN, 2017