NASA Space Grant 2017 8.21 Eclipse Ballooning Project

MSGC

RFD 900MHz Still Image Payload

Tristan Running Crane
Advised by: Berk Knighton and Randy Larimer

Abstract

As part of the NASA Space Grant 2017 Eclipse Ballooning Project, Montana Space Grant Consortium designed a camera system capable of transmitting still images from a balloon borne payload to a remote ground station via a RFD-900 radio. This payload consists of a Raspberry Pi, a Raspberry Pi camera, custom power supply and a RFD 900MHz radio. This system was designed to record pictures at high and low resolution at altitude. The low resolution images could be selected for downlinking back to the ground via the radio, while the high resolution pictures were saved to the Raspberry Pi memory card. This poster describes the full hardware configuration, communication protocol, and graphical user interface used to receive the transmitted images.

Hardware Configuration

The Still Image Payload is powered by two 3.7V 6600mAh batteries which are connected to the custom power supply. The board supplies power to the Raspberry Pi and RFD 900MHz Radio when the key switch is in the on position. This powers board which supplies power to the Raspberry Pi through its GPIO pins and to the RFD Radio through its pins. The rest of the components on the Still Image Payload are powered through the Pi's GPIO pins such the real time clock, and the motor for adjusting the position of the Raspberry Pi camera. The camera is connected and powered to the Pi through a camera serial interface (CSI) port on the Pi. The RFD 900MHz Radio is connected to the Pi through its GPIO pins and communicates over serial. The RFD has two quarter wave Monopole antennas, which are connected to a ground plate, on the bottom of the payload which are configured to be perpendicular to each other. This allows for communication between the ground and the payload midflight. The RFD and the two antennas are connected with a coaxial cable.

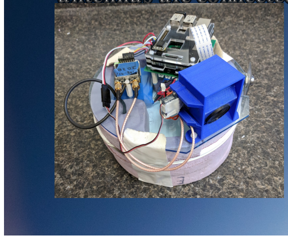


Figure 1. The Still Image Payload, a custom power board, Raspberry Pi, Pi Camera, and RFD 900MHz Radio

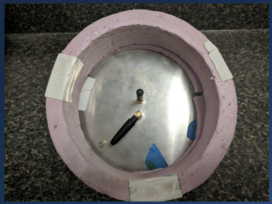


Figure 2. The Still Image Payload, antennas pointing downward connected to a ground plate

Communication Protocol

The Raspberry Pi communicates to the RFD 900 MHz radio through serial and the RFD radios communicate to each other through what is essentially wireless serial. The Raspberry Pi runs a python program midflight which takes a picture a high and low resolution picture and allows for the transmitting the picture mid flight.

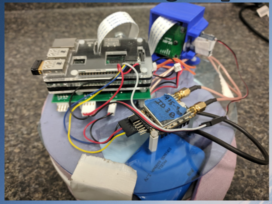
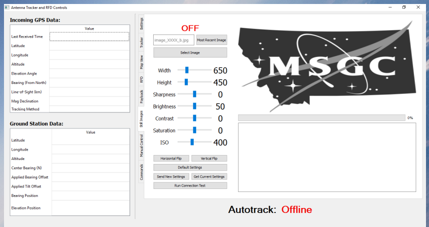


Figure 3. The RFD 900 MHz radio connected to the Raspberry Pi and the power supply

When requesting a picture the ground station sends an serial command which is transmitted between the radios and then given to the Pi which interprets this byte as a command to send the picture through the RFD radio. How the Pi sends the picture to the radio is it first breaks the picture down each pixel into 5 bytes which are Red, Green, Blue, X-position, and Y-Position. The Pi then sends these values in binary to the radio in the form of packets, which is 7000 bits, then sends the packets to the other radio and is received by the ground station. The ground station then reassembles the picture using those RGBXY values. However this process is not error tolerant, the program creates a hash value for each packet of data and is used as a checksum to determine if there are errors in the data being received. While this allows for an uncorrupted picture being sent this also means that if one bit of data in your packet is wrong the entire packet has to be resent and can cause the receive time to increase depending on noise level and other environmental effects.



Antenna Tracker and RFD Controls

Incoming GPS Data:

	Value
Last Received Time	
Latitude	
Longitude	
Altitude	
Elevation Angle	
Bearing (From North)	
Line-of-Sight (km)	
Mag Declination	
Tracking Method	

Ground Station Data:

	Value
Latitude	
Longitude	
Altitude	
Center Bearing (N)	
Applied Bearing Offset	
Applied Tilt Offset	
Bearing Position	
Elevation Position	

OFF

image_XXXX_b.jpg

Width 650
Height 450
Sharpness 0
Brightness 50
Contrast 0
Saturation 0
ISO 400

0%

Autotrack: Offline

Figure 3. The Graphical User Interface for the Still Image Payload

Graphical User Interface

The Graphical User Interface (GUI) for the Still Image System allows for requesting and modifying pictures from the Ground Station while the payload is at altitude. How the GUI modifies the parameters for the pictures it will take in the future is first adjusting the sliders in the GUI such as resolution, sharpness, brightness, contrast, saturation, and ISO to the desired values then sending a command to the payload. If the payload receives the command it then sends back an acknowledge the GUI then sends the data which is received by the payload the program running then takes this data and modifies itself. If it does not receive the initial command or the data then nothing is changed and the program continues to run. How the GUI requests a picture is very similar to modifying the parameters of the picture. The GUI sends a command which is recognized by the Pi and sends back packets of data of the most recent picture taken.

Future Work

In the future I would like to continue working on the Still Image Payload and would like to integrate sensors onto the Raspberry Pi on a custom board which would allow for recording atmospheric data, such as pressure, GPS, both internal and external temperature. I would like to make the system error tolerant and reduce the time to receive pictures.

References

David Schwehr – The original designer and coder of the Still Image Payload

Trevor Gahl – Also worked on Still Image Payload code

Skylar Tamke – Designer of the custom Power Supply