Brittany Cahill, Patrick Cremin, Marykate Drennan, Chris Susco, & Javier Rivera Gonzalez

EE 41440 Senior Design

21 February 2021

Design Review 1

I. Detailed Design of Each Subsystem

The first major subsystem is the orientation correction subsystem, which consists of a 3.7 V LiPo battery, 5 V boost converter, Itsy Bitsy microcontroller, 9DOF orientation sensor, and three servo motors. The orientation system will independently control its three landing legs using servos and lead screws. The servos being used are 2000 Series Dual Mode Servo, running on continuous rotation mode. The servos will be controlled by an Arduino Nano board. The system will first receive a landing detection serial input from the Planetary Landing System (PLS) Raspberry Pi. Once this signal is received, the Arduino will use angular position data and loop through a proportional control algorithm to level the lander. The algorithm used to level the body of the lander will use the three legs to orient upon two primary axes of rotation. The X and Y axes of rotation correspond with the primary axes of the PLS sensor. Upon landing, the three legs of the lander will all be perpendicular to the payload body. In order to orient the lander on the two axes, the orientation sensor will continuously measure the angle and adjust all three legs until the payload is within five degrees of vertical with margin for error.

The second major subsystem is the imaging subsystem, which consists of a 3.7 V battery, 5 V boost converter, Raspberry Pi, transmitter, camera adapter, and four cameras. The team will be utilizing Raspberry Pi cameras with wide-angle lenses. The cameras will be offset 90 degrees from one another. With a horizontal field of view of approximately 120 degrees, the cameras will be able to capture a 360-degree image by stitching together four pictures after they are transmitted to the host computer. The Raspberry Pi camera and wide angle lens are shown in Figure 1. These cameras will be connected utilizing a ArduCam Multi Camera Module. This module is shown in Figure 2. This will allow the team to take a picture with each individual camera. The images taken are stored in a .jpg file format, which will allow for easy manipulation and processing later on. After the images are taken, they will be stitched together utilizing a Python OpenCV algorithm. This will generate the desired 360-degree image file, which will be sent through the data transmission subsystem. Because the Raspberry Pi utilizes the Linux kernel and has a file system, the four initial images and the final panoramic image will be able to be stored in flash memory for easy accessibility.

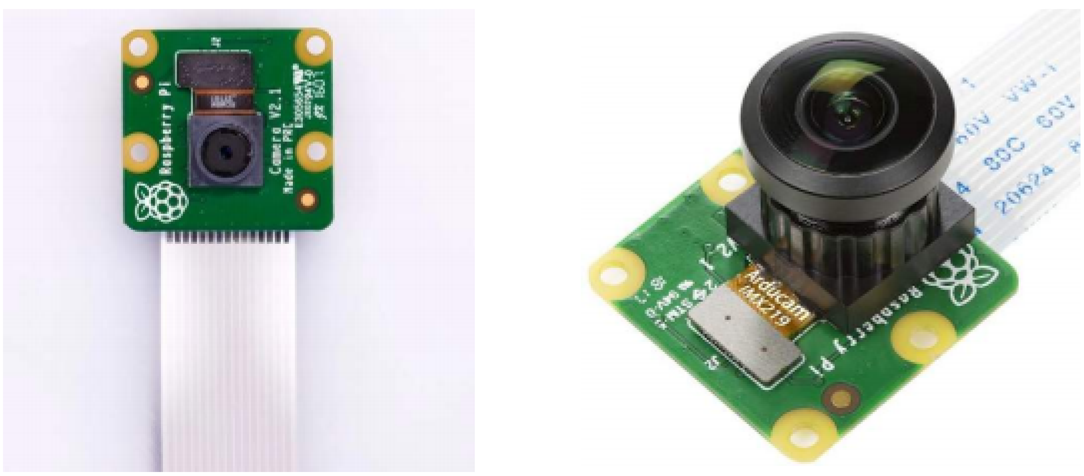


Figure 1. Raspberry Pi camera (left) and wide angle lens (right)

Figure 2. ArduCam Multi Camera Module

The third and final subsystem is the transmission subsystem, which consists of a receiver and the second Raspberry Pi. One Raspberry Pi will reside in the PLS, while this Raspberry Pi will be located at the launch site. The Raspberry Pi in the PLS will act as a transmitter, while the Raspberry Pi at the launch site will act as a receiver. This system will be powered by a 3.7 V battery connected to a 5 V boost converter, allowing the payload subsystems to be powered for at least 2 hours on the launch pad. The transmission itself will be handled by a pre-packaged radio transceiver developed by Adafruit and the LoRa Radio Bonnet RFM96W. The radio bonnet is shown in Figure 3. This device will allow the team to very easily transmit and receive data using a pre-packaged Adafruit API. The data will be transmitted using the LoRa scheme over a 433 MHz carrier wave, which can carry a signal over 2 km. This distance varies with antenna schemes, so the team will need to test to determine the distance at which the transceivers lose sight of one another. The distance of 2 km was measured by Adafruit using an omnidirectional antenna, so the switch to a dipole antenna should increase the range of transmission. This device also allows for handshaking and confirmation signals, to ensure that the data packets were received properly and mitigate data loss. The image will need to be split into smaller packets in order to be transmitted over the system, which will be done using OpenCV. Furthermore, this device will allow for easy debugging on an LCD, and includes push-buttons. Each of these devices will be connected to a dipole antenna, which was chosen due to the consistent orientation of the subsystem.

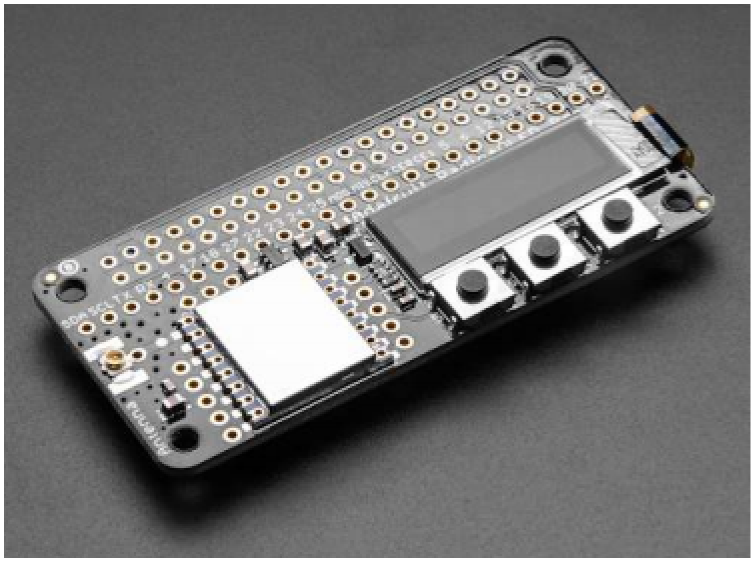


Figure 3. Adafruit LoRa Radio Bonnet RFM96W

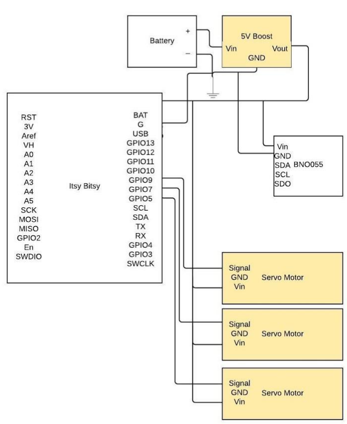
II. Description & Function of Major Components

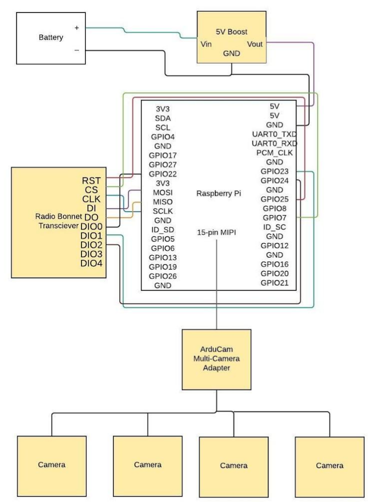
1. The orientation and imaging subsystems each contain a battery and 5 V boost converter. Each of these components plays a vital role in ensuring that the systems can be powered for two hours on the launch pad, in accordance with NASA requirements.
2. The Itsy Bitsy is connected to three servo motors and the orientation sensor, which not only allows the orientation system to autonomously orient the PLS upon landing to within five degrees of vertical, but also sends a confirmation via serial communication.
3. The Raspberry Pi within the imaging subsystem is connected to a transmitter, which will allow it to send the orientation system's confirmation of success to the receiver, as well as the panoramic photo.
4. This Raspberry Pi is also connected to the camera adapter, which is in turn connected to the four Raspberry Pi cameras. The cameras will collectively take four separate photos of the landing site. The adapter will stitch these photos together into a 360-degree image so that the Raspberry Pi can send the finalized panorama.

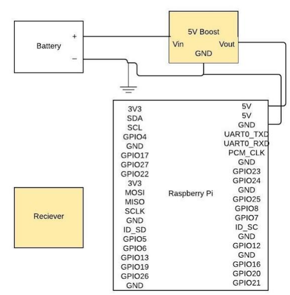
5. The transmitter is wirelessly connected to the receiver. The receiver and second Raspberry Pi make up the transmission subsystem. While connected to the host computer, they can receive the orientation and photographic data from the transmitter.

III. Essential Connections

All components of the system will be powered by a 3.7 V battery and a 5 V boost converter. As a result, all subsystems will be powered by 5 V and 1 A, which is the required amount of power for Raspberry Pi's. Essential connections are shown in the pin-out diagram below:







IV. Problems & Action Plan

We will likely have problems arise when we are writing the remaining code for the system. Fortunately, Raspberry Pi code references can be found from a number of sources online, as well as from last year's team documentation, so our action plan is to begin solving problems by consulting these resources.

V. Team Site

The proposal and high level design documents have been uploaded to the team web site.