Johns Hopkins University

Project 8

Quad Copter downloading IMU RPi to Host over WiFi

Miles Gapcynski

EN.605.715.81.FA19 - Software Development for Real-Time Systems

Professor Doug Ferguson

11/17/2019

Contents

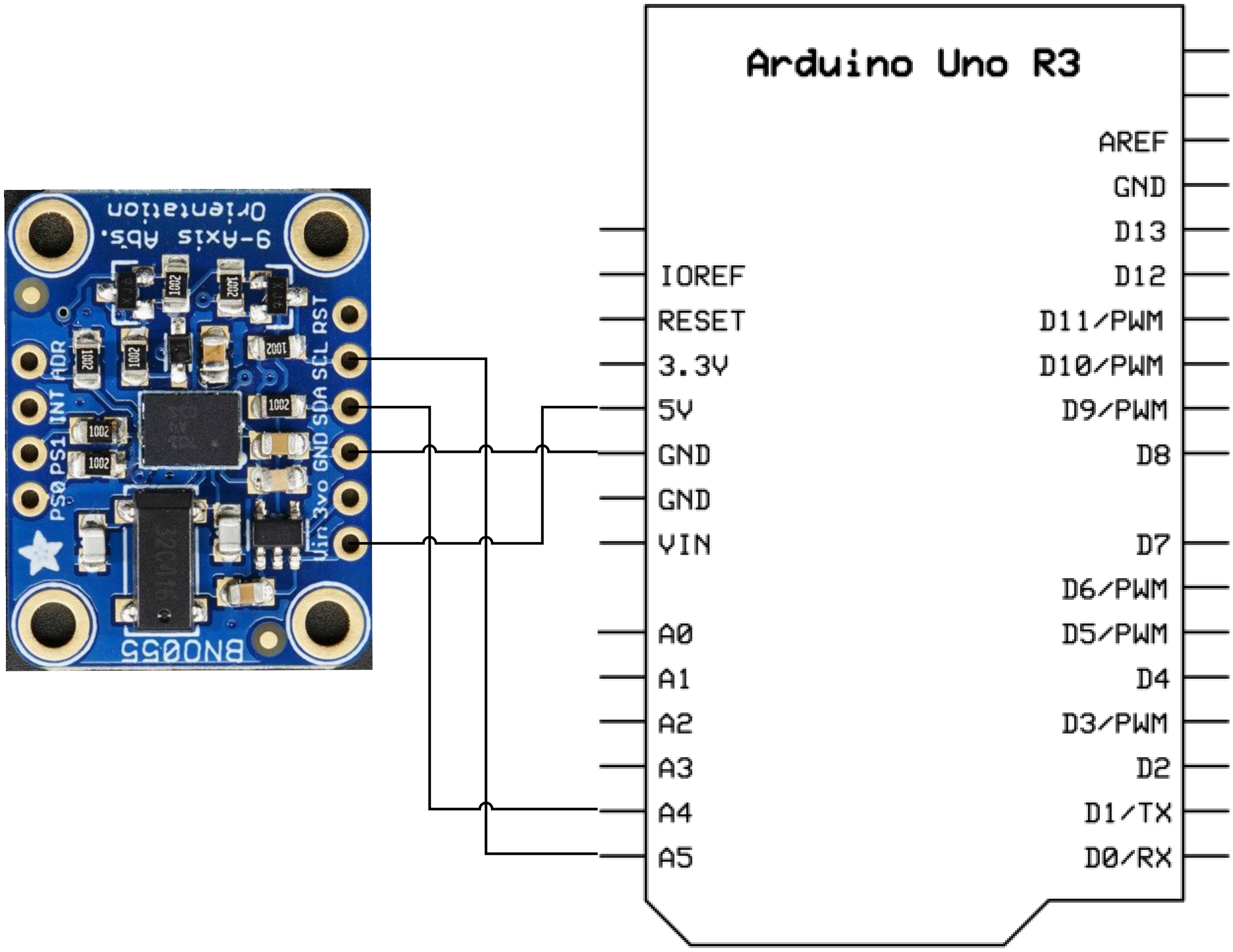
Derived Requirements

The following requirements were derived from the Project 8 Quad Copter Downloading IMU RPiToHostWiFi document:

- The Arduino shall periodically read orientation data from a BNO055 IMU sensor connected via an I2C bus.
- The Arduino shall transmit the orientation data as comma-separated roll, pitch, and yaw (RPY) values over a serial connection (USB).
- The Raspberry Pi shall periodically receive the comma-separated RPY values from the Arduino over a serial connection (USB).
- The Raspberry Pi shall transmit the RPY data over a socket to a host machine.
- The Raspberry Pi shall act as a wireless access point so that the host machine can connect to the Raspberry Pi over WiFi without a router.

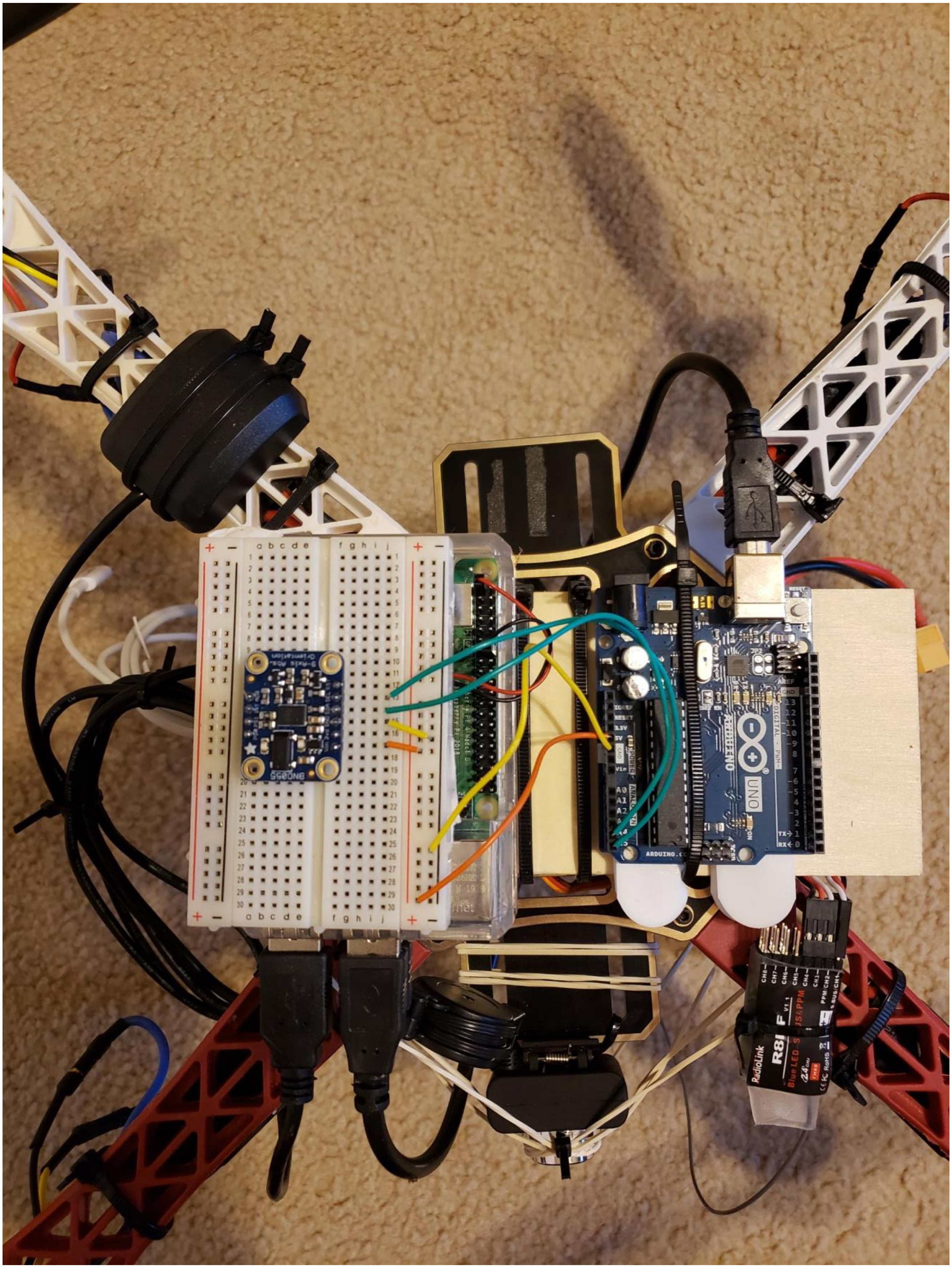
Hardware Design

The following diagram is a schematic of the circuit connected to an Arduino Uno (rev. 3) that receives orientation data from a BNO055 IMU sensor using an I2C bus. The orientation values, which are Euler angles, are read at a rate of 10 Hz and transmitted over serial (USB).

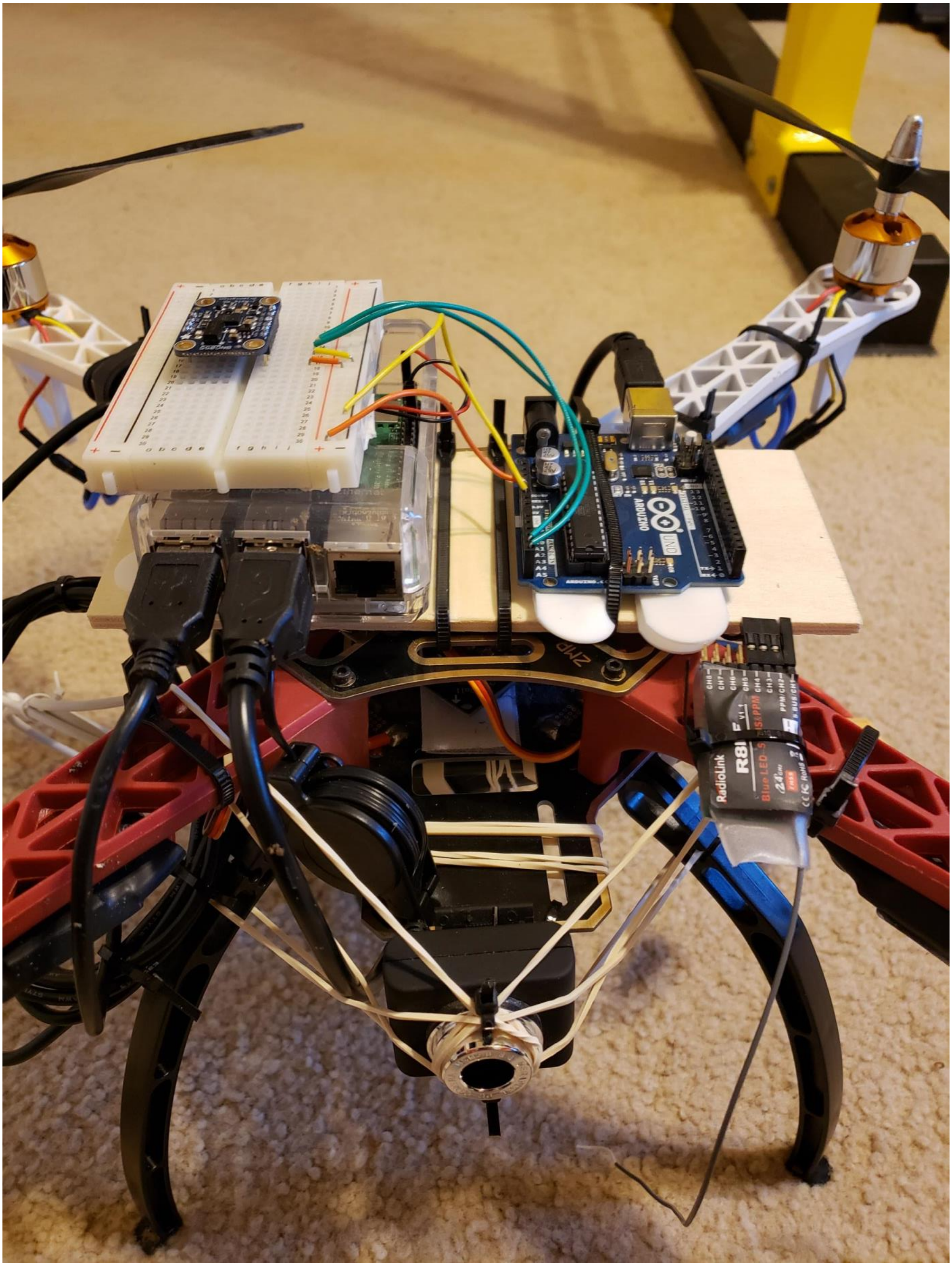


QuadCopter and Board Layout

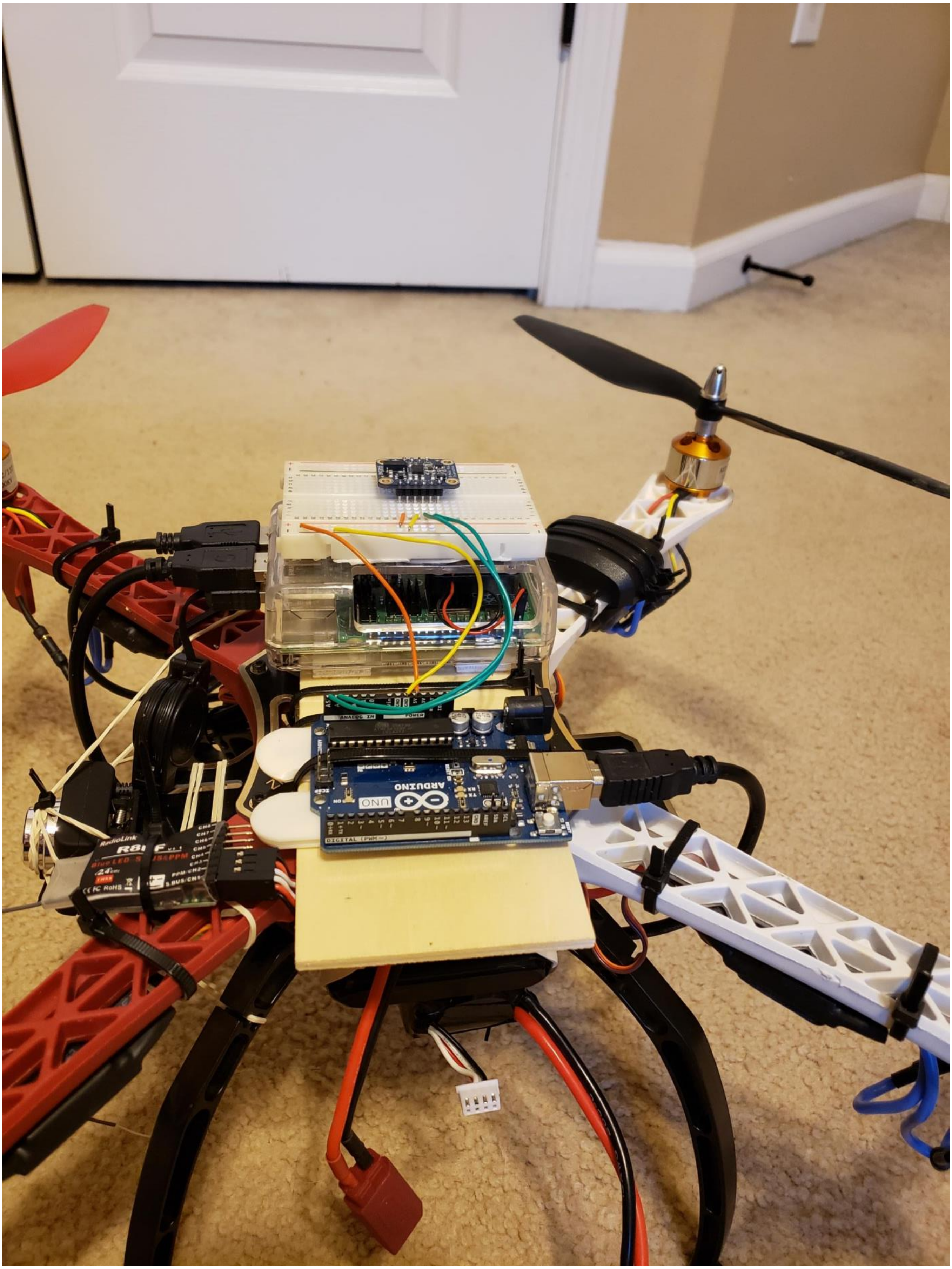
Top view of how the Raspberry Pi, Arduino, and BNO055 IMU sensor are connected and attached to the Quadcopter:



Front view of how the Raspberry Pi, Arduino, and BNO055 IMU sensor are connected and attached to the Quadcopter:



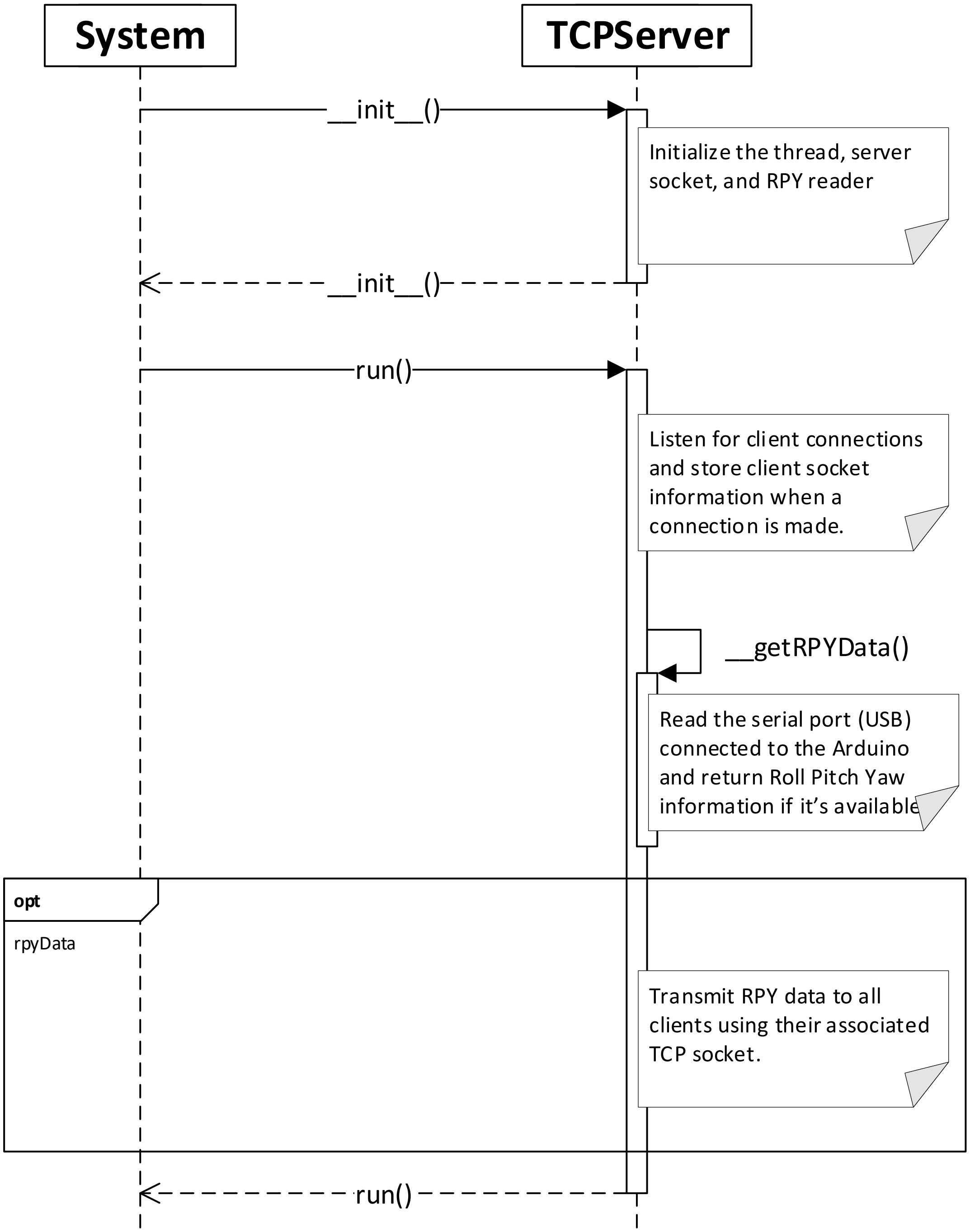
Side view of how the Raspberry Pi, Arduino, and BNO055 IMU sensor are connected and attached to the Quadcopter:



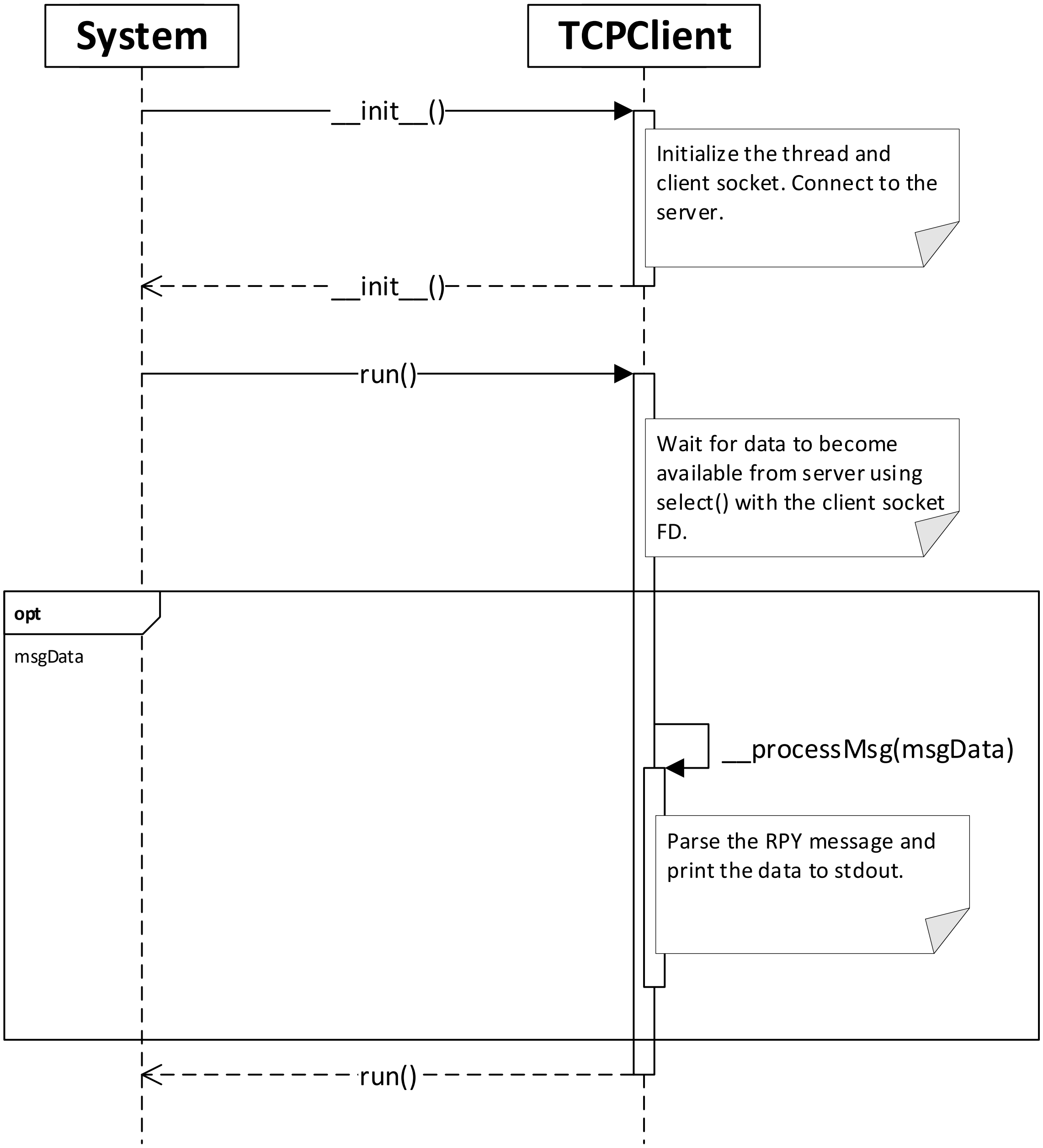
Software Design and Implementation

Sequence Diagrams

The following diagram is a sequence diagram of the server that reads and transmits RPY data to a client:


System
TCPServer
__init__()
Initialize the thread, server socket, and RPY reader
__init__()
run()
Listen for client connections and store client socket information when a connection is made.
__getRPYData()
Read the serial port (USB) connected to the Arduino and return Roll Pitch Yaw information if it’s available
opt
rpyData
Transmit RPY data to all clients using their associated TCP socket.
run()

The following diagram is a sequence diagram of the client that receives the RPY data from the server and prints the information to stdout:


System
TCPClient
__init__()
Initialize the thread and client socket. Connect to the server.
__init__()
run()
Wait for data to become available from server using select() with the client socket FD.
opt
msgData
__processMsg(msgData)
Parse the RPY message and print the data to stdout.
run()

Video Demonstration

The following video demonstrates the Raspberry Pi streaming RPY data from the quadcopter to a host machine. The video starts off with a brief description of the project, followed by a demonstration of the roll, pitch, and yaw data being displayed on the host machine while the vehicle is rotated by hand. The video shows that same manual demonstration from a separate camera, followed by the actual flight.

The RPY data was lost during the actual flight, as an exception occurred reading from the serial port. I had not encountered this problem during development or testing, and I unfortunately was not able to record another successful flight as I ran into communication problems between the remote control and receiver/flight control board. I tried disconnecting and reconnecting cables to ensure there was good contact between the pins and cables, but that didn’t resolve the communication issues. I suspect the pins from the receiver to the flight controller board are not making very good contact, but it’s possible there’s some interference going on. I also continue to have intermittent connection issues between the Raspberry Pi and host machine, which I attribute to a weak WiFi signal that quickly gets worse as the quadcopter gets further away from the host machine.

Flight

https://www.youtube.com/watch?v=tVA5oIGiHUk

https://www.dropbox.com/s/4nts4ou0y0tam8t/Miles_Gapcynski_EN_605_715_81_Project_8.mp4?dl=0