* Realized CLK and MISO pins were shorted after probing SD card module pins with the DMM’s continuity test
  + Fairly easy fix as there was wire sticking up through one of the protoboards from the CLK pins touching the MISO rail
  + I cut those wires, probed CLK and MISO again and saw no signs of a short
* I was able to connect to the servos, RGB LED, and buzzer through the flight computer, but ran into issues with all devices using SPI communication
  + I first noticed this issue trying to initialize the 1st BNO085 sensor where the serial terminal would freeze after “starting BNO085 test”.
  + I thought initially it was an issue with the serial terminal and the usb cable not allowing transfer of data to it, but I was able to resolve the serial terminal issue by restarting the teensy and matching the baud rates of my code files. However, it was still freezing.
  + I realized, the microcontroller was either freezing or crashing when I called BNO085.begin\_SPI(BNO085\_CS), suggesting something was preventing SPI communication from initializing
  + The most common reason for this is incorrect wiring of the pins, pins disconnected, or pins being shorted. To figure out if this was the case I probed all pins with my multimeter to verify they were corrected to the correct teensy pins. I also probed the pins within the BNO085 board to ensure there wasn’t any shorts. After doing this all wiring seemed to be correct and there weren’t any signs of shorts
  + It also could have been an issue with improper voltage supply and grounding, but I probed between the modules ground pins and vin pins and got a voltage of 3.3V, which is correct
  + I also tried other boards like the 2nd BNO085 and the BMP388 and ran into the same exact issue, which suggests this is very likely an SPI communication issue (which I already knew)
  + Because of the fact that the wiring appeared correct, there were a couple of issues I had considered:
    - There could be bus contention issues where multiple devices are trying to drive the miso line. I changed my code to pull all cs pins high except for the one I’m trying to communicate with, which didn’t fix anything (I figured this would be the case as Adafruit’s breakout boards have an internal pullup resistor on each of the CS pins).
    - There are probably one or multiple SPI devices interfering with MISO. Other people have had issues especially with SD modules and IMU sensors that cause conflict on the SPI bus. To debug this, I will disconnect all MISO lines except for the one I want to try and connect to and see if this is the issue
    - This could also just be an issue of the wires being connected enough to appear as continuous on the multimeter, but not enough for continuous operation of the modules

All of these issues were resolved by switching to a PCB and changing most of the SPI communication to I2C. Although slightly slower, the simplicity of I2C makes it much less prone to errors and bus conflicts.