

## ME 410 - Week 1 Summary

Logan Boswell

### Summary

For this week, the main goals/tasks were to set up the Raspberry Pi and IMU so that Pi is able continuously receive gyro, pitch angle, and roll angle data which will be used later for flight control. To start, my partner and I connected to the Pi over SSH after connecting to the Pi's WiFi. From there, we created the flightControl directory and copied over the starter code from canvas so we could look over the code and see what needed to be added. In order to obtain readings for the IMU data, we found (using the datasheet) and added the 6 necessary registers to the `wiringPiReadI2CReg16()` function calls so the raw accelerometer and gyro data for all 3 directions could be read. We also edited the `wiringPiWriteI2CReg8()` function calls to set the range of gyro values to  $\pm 1000$  deg/s and the range of accelerometer values to  $\pm 3$ g. Conversions were determined for both the gyro and accelerometer readings (based on the range of values for the gyro data, accelerometer data, and counts) so the raw data could be converted to proper units (deg/s and gs) and be saved in an array. After confirming that we were correctly receiving data from the IMU, we added functions to calculate the roll and pitch angles by using `atan2()` with accelerometer data (`acc_y`, `acc_x` for pitch and `acc_z`, `acc_x` for roll). With these functions written, the `calibrate_IMU()` function could be edited to calculate offsets when the code is run before the main loop starts. Averages for each of the 3 directions of the gyro data, the roll angle, and the pitch angle were calculated over 1000 samples and saved as global variables to be used as offsets. From here, we confirmed that our gyro, roll angle, and pitch angle data was correct and correctly updating so we could get checked off.

### Assessment - What Went Well

Overall, I feel like this week's tasks went pretty smoothly. My partner and I both have previous experience working with I2C communication and IMUs so we were familiar with how to read from/write to the device and only needed to look up the specific registers necessary for our tasks.

### **Assessment - What Did Not Go Well**

We had two minor issues during this week: forgetting to subtract the calculated offsets and having the wrong sign for our gyro data about its z-axis. Both of these were not too serious and only required small changes in our code to make them work properly. Also, I feel like we could use a better way to edit our code on the Pi. Since we had accessed the Pi over ssh, we were just using the nano text editor in the terminal, but in future weeks I think it would be better to find a different way to edit code on the Pi.

### **Assessment - Adjustments for Next Class**

Overall, I think the main adjustment we need to make is just finding a better process for editing the code on the Pi. While editing in nano does work, I feel like it definitely slows our progress.

### **Team Member Effort**

Me - 55% (Edited the code on my laptop)

Ben - 45% (Pulled up datasheet/slides/whatever else we needed on this laptop)