**Software System Design**

Cornell Hyperloop’s pod contains one embedded ODROID-C2 computer running the Ubuntu operating system, four Arduino Uno microcontroller boards each housing an ATmega328P microcontroller, and one Arduino Mega microcontroller board housing an ATmega2560 microcontroller. The four Arduino Unos are contained in the Fudicial Detector, Brake/Low-Speed module, and Front and Rear analog modules. The Arduino Mega was utilized in the Flight Control module due to the processing demands of its operationally critical task. These five modules communicate with the pod computer using USB connections carrying 100kbaud serial data, and the pod computer communicates with the pilot laptop over WiFi with an ESP8266 WiFi module. The ODROID utilizes information from the network state, Arduinos, and pilot computer to autonomously determine and control the state and operation of the pod.

**Definition of modules:**

* The Fudicial Detector (FD) utilizes an optical sensor to detect the occurrence of fudicial markings along the tunnel. These marks are 4” wide and occur at 100 *ft* intervals. Assuming a relative pod speed of 110 *m/sec*, the marks will be measurable for 400 *μs*. Therefore, the optical sensor must have a minimum sampling rate of 5 *kHz*.
* The Brake/Low-Speed Module manages all actuation of the pod. State information from the hydraulic system controlling the brakes and the motor controllers associated with the low-speed system are sent to the ODROID for analysis. Brake caliper positions are controlled by an inner-loop controller in this module.
* The Front and Rear Analog modules communicate with a range of sensors (thermistor, potentiometer, accelerometer, etc.) across the pod’s suspension, lateral control components, and power supplies.
* The Flight Control Module communicates with the IMU and the Fudicial Detector to determine the current physical state of the pod. The output of the IMU is used to estimate the pod’s velocity, orientation, and position. These estimates are utilized in conjunction with information from the FD to filter fudicial marker detections in an Extended Kalman Filter. The resulting estimates are then passed through a trajectory-tracking microcontroller to generate brake commands for the pod.