**Hyperloop Controls and Communications System**

**Shternshain, Bar (BS Electrical Engineering)**  
**Smith, Sean (BS Electrical Engineering)**  
**Tang, David (BS Electrical Engineering)**  
**OUSHEROVITCH, DANIEL (BS Electrical Engineering)**

---

**Introduction**

Hyperloop is a new mode of transportation that involves a passenger-pod traveling in a low pressurized tube [1]. The task of our project was to design a communication system that uses a microcontroller and sensors to read telemetry and safety data throughout the pod. This data includes temperature, battery pressure, brake operation, and speed estimation. The communication system for the pod used two types of protocols, Control Area Network (CAN) and Inter-Integrated Circuit (I2C). These protocols helped relay the data taken by the sensors to the microcontroller with the help of various Application programming interfaces (APIs) designed by our team.

In addition, it was also our responsibility to control the pod motor and battery management systems over the CAN bus. The motor was controlled by converting an acceleration profile into CAN commands for the electronic speed control. The battery management systems were also interfaced with over the CAN bus in the same way that car’s on board computer interfaces with its sensors. These protocols helped relay the car’s on board computer interfaces with its sensors.

**Methodology**

The following diagram shows the approximate physical location of various devices within the pod and how they are connected with the MCU. Not the various types of transmission methods.

For the I2C bus, each sensor data sheet was studied in order to determine the most efficient operating conditions for the sensor. For the MLX temperature sensors, the internal limitations had their network operate on the 100 kHz bus. In addition, due to the physical distance limitations, an additional buffer amplifier was placed on the bus.

The piezoelectric air pressure sensors and the time of flight style optical braking sensors operate on the separate 400KHz bus. Part of the reason for this is that air pressure loss in a battery compartment is a catastrophic event that if much more hazardous to a pod than a slow rise in battery temperature. Brake application is also a quick discrete event which benefits from faster sensor response time.

The second concern of our team was the sensing of simple GPIO analog and digital peripherals. Interrupt service routines were designed for the retro reflective sensors to count strips on the inside of the vacuum tube. The function of these sensors was to confirm the position of the pod. Analog brake line pressure sensing was accomplished with a current based pressure sensor, a ballast resistor, and a simple ADC block on the MCU.

**Analysis and Results**

Our first effort was to send CAN frames over the 100KHz bus [5]. Though the signal appears a bit noisy, this is just an artefact of probe impedance. A receiver node was able to take commands from the sender and send frames back and desired functions.

Our second goal was to assemble a multi-sensor, fire- and forget style I2C bus for non-blocking sensor reads. We were able to achieve this, running multiple temperature sensors on the same bus at just under both of their maximum frequencies. This was confirmed by their data sheet limitations.

**Summary/Conclusions**

In terms of data management, we implemented some DSP running average filters, a data structure for telemetry storage, and began work on storage of various motor acceleration profiles.

By using CAN, I2C, and various GPIO peripherals we were able to design a system covers the safety and competition requirement laid out by SpaceX and will help the Spartan Hyperloop team moving forward with their full pod design.

Due to the current situation, full physical implementation was not achieved, but we have paid great care to design with electrical interference, other subsystems, integrability, and modularity in mind.

Moving forward we would like to see more RTOS style software to manage peripherals and invest time in pod state machine decision making algorithms.

---

**Key References**


**Acknowledgements**

Special thanks to the Spartan Hyperloop team and to our advisor Dr. Ping Hsu.