

# TDPS Navigation Module Design Notes

---

Sidi Liang, 2023

## Patio 1

---

### Design Goal:

The vehicle should be able to follow the brick line and precisely drive through the bridge and the gate.

### Possible Design

#### Plan A: Pure CV

##### Hardware Involved:

- OpenMV
- Ultrasonic sensor (installed facing the front)(up to 2)
- Optional: AprilTag Beacon

##### Comm. Involved:

- I2C with the Ultrasonic sensor
- UART with the master control

Using OpenMV, apply filter to obtain the boundary of the brick line and use linear regression to calculate the target line to follow. Use the Ultrasonic sensor to determine when to turn right for the bridge and when to turn left for the gate. AprilTag is a kind of printed tag that can achieve localization when scanned by the camera. If ultrasonic sensor alone is not sufficient for the vehicle to precisely align with the bridge, the AprilTag Beacon maybe placed to assist the localization. On the microcontroller of OpenMV, calculate the difference between the target line and the current line, and calculate the amount of steering with PID. This steering command should be sent to the master control in real time with UART.

##### Potential issues:

- Difficulties with line tracking
- Difficulties aligning with the bridge

#### Plan B: IMU assisted CV

##### Hardware Involved:

- OpenMV
- Ultrasonic sensor (up to 1)
- BMI160 IMU sensor (up to 1) (Alternative: That supported out-of-the-box by OpenMV but the very expensive (~20 usd) one (I forgot what it is called, but, you get the point))
- Optional: AprilTag Beacon

##### Comm. Involved:

- I2C with the Ultrasonic sensor
- I2C with the IMU sensor
- UART with the master control

Using OpenMV, apply filter to obtain the boundary of the brick line and use linear regression to calculate the target line to follow. Use the IMU data as a complement and verification of the CV results.

Use the Ultrasonic sensor (or AprilTag) to determine when to turn right for the bridge and when to turn left for the gate. On the microcontroller of OpenMV, calculate the difference between the target line and the current line, and calculate the amount of steering with PID. This steering command should be sent to the master control in real time with UART.

**Potential issues:**

- Difficulties with the implementation of the IMU sensor
- The performance of OpenMV cannot satisfy the need
- Occupying too much pins (could this be solved by adding extension boards?)