Car Obstacle Detection & Rerouting

Ahmed Abdelrahman	900161054
Nouran Flaifel	900160793
Zeyad Zaki	900160268

Agenda

Recap

Interfacing with accelerometer

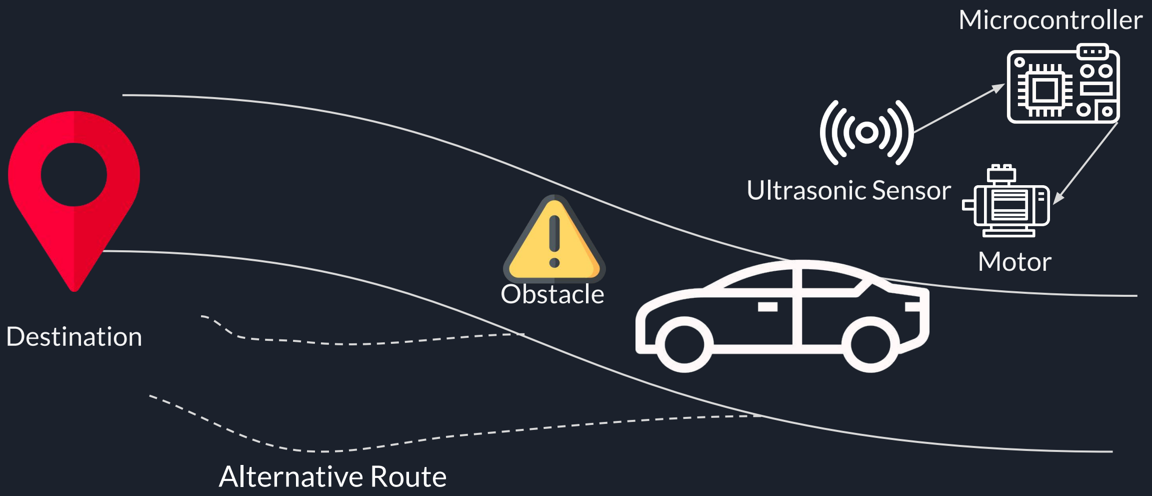
Prototype Features

Prototype Demo

What’s Next

Recap

The smart car will be moving in a specific track to reach a final destination. It should be able to detect any obstacles that it faces and be able to reroute in order to avoid them while still aiming to reach the destination.



Recap - Hardware Resources

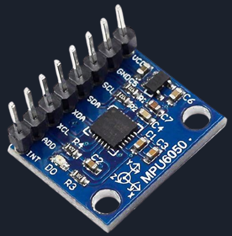
01 STM32 Microcontroller

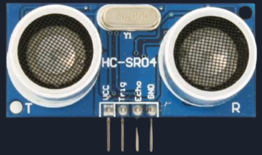
02 Dagu Wild Thumper 4WD Chassis

03 Tri-axial Accelerometer (MPU6050)

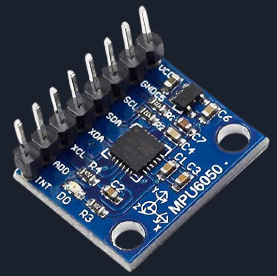
04 Pololu TReX DC Motor Controller

05 3 HC-SR04 ultrasonic ranging module





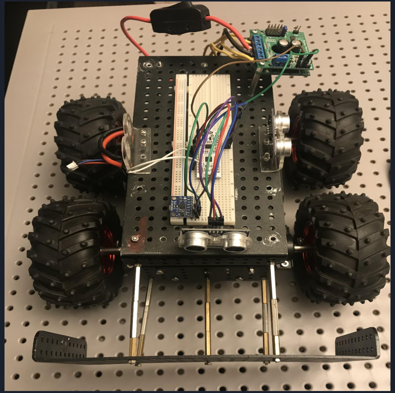
Interfacing with MPU6050

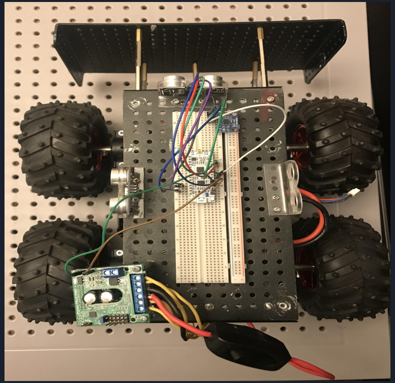


- **I2C** is used as the communication protocol between the 2 devices.
  - Master: STM32L4.
  - Slave: MPU6050.
- Communication takes place through these two wires namely:
  - **Serial Clock (SCL)**: for sharing the clock signal generated by the master with the slave. -I2C is synchronous-
  - **Serial Data (SDA):** Sends and receives data between the master and the slave.
- We used “**MPU6050 6 axis library**” that uses HAL library to get the basic device driver functions which were used to get the accelerometer and gyroscope raw and scaled readings.

Prototype Features

- We’ve used the car design shown to best fit our requirements
- There is a place for 3 ultrasonic sensors but in the milestone we used only the front one
- In this prototype the car moves and whenever it detects a front obstacles it turn right to avoid it
- At the same time we printed the distances received from the ultrasonic sensor along with the accelerometer readings for visualization





Prototype Demo

What’s Next

- Using accelerometer readings to detect the device’s current location
- Creating a map for the device with starting and ending point
- Rerouting using the shortest possible distance

Thank you!