Iowa State University
College of Engineering

Race of Doom

Team sdmay24-43 Andrew Kraft - Testing, Circuit Design Jack Doe - Project Manager;
Gavin Petrak - Team Organization Jacob Nedder - Testing, Team Coordination Peter Wissman - Client Interaction
Client: Timothy Bigelow Advisor: Timothy Bigelow

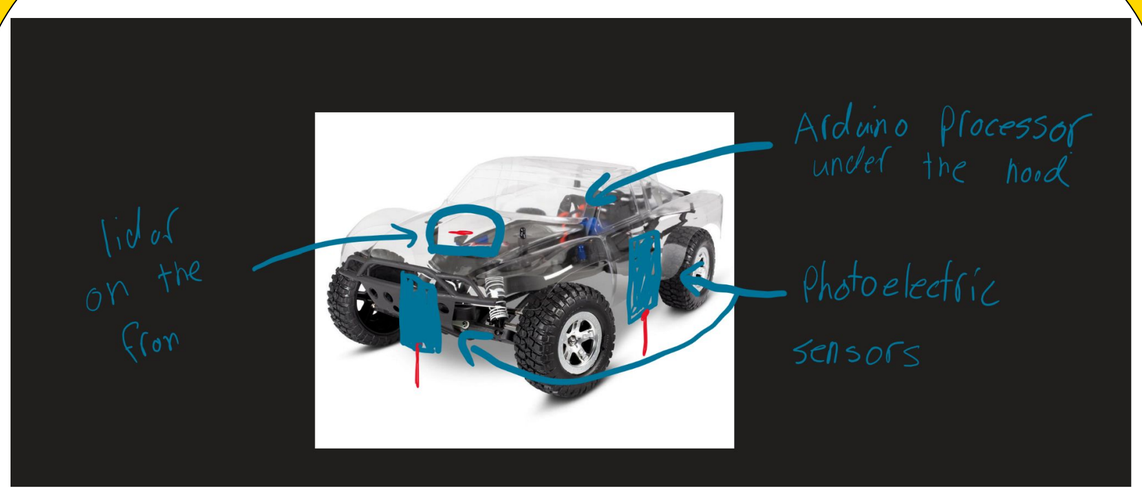
Problem:
The Race of Doom challenges the car development teams to design autonomous vehicles with onboard resilience to crashes and other hazardous obstacles.

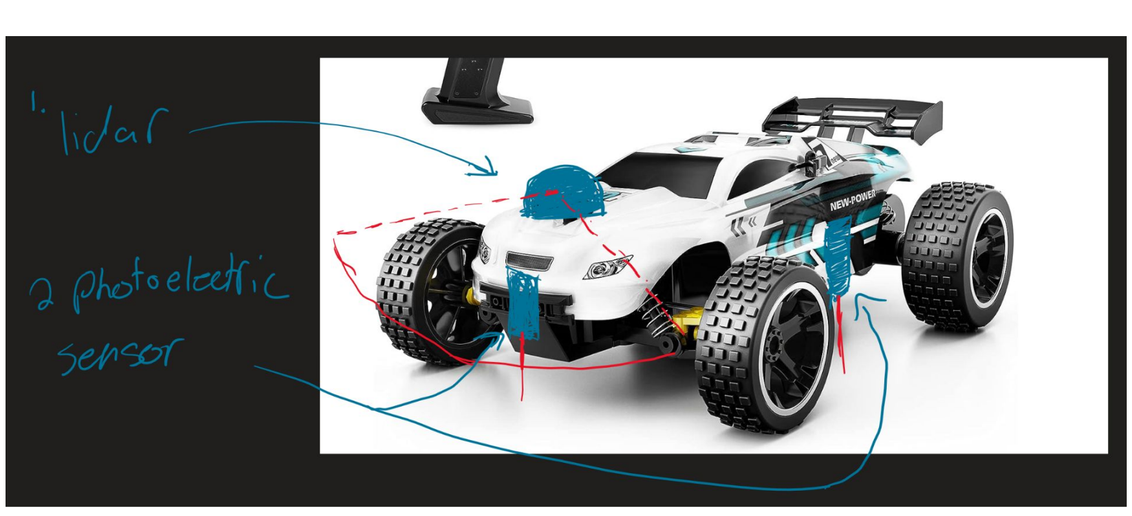
Intended Users/Usage:
The RC Car developed for this project is intended for racing, navigating a track containing dangerous obstacles intended to impede progress. A single driver is allowed, controlling only the car's acceleration.

Operating Environment:
The environment for this project was designed by the Track Development Team, featuring turns, a ramp, a tunnel, and autonomously moving doorways.

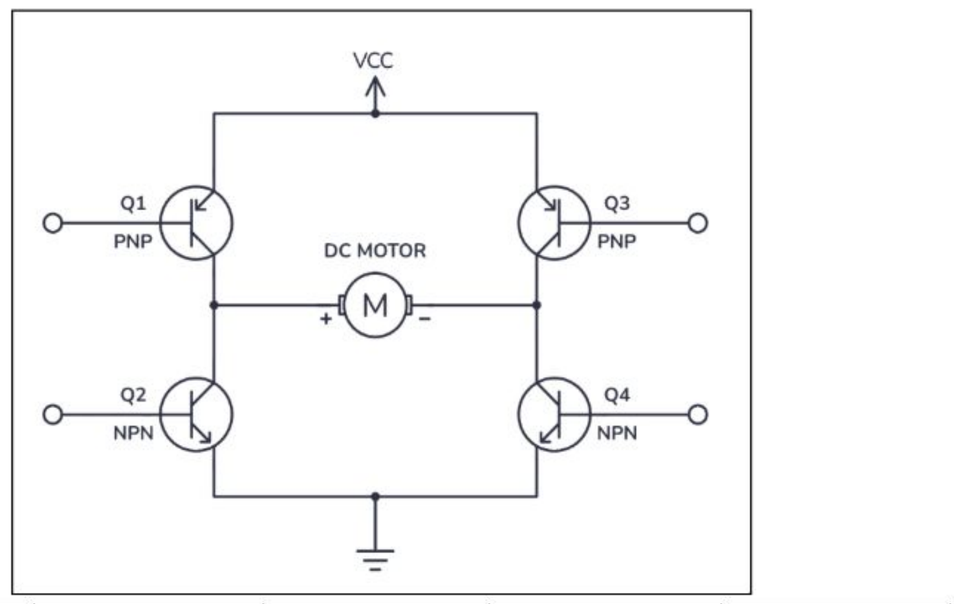
Design Requirements
- The RC Car shall autonomously steer away from obstacles on the track
- The RC Car shall stay within the bounds of the track
- The RC Car shall be protected from the track-hacking source
- The Driver shall control only the speed and acceleration of the RC Car

Design Constraints
- The RC Car must cost less than $500 to build, modify, and develop.
- The RC Car should be able to sense its environment using only sensors installed on the vehicle.
- Each obstacle for the RC Car should be overcome as quickly as possible to win the race.





Figures 1 & 2: Initial designs for the RC Car



	GPIO/BCM Pin	Turn Left	Turn Right	Straight
Q1 - PNP	13/27	1 - Off	0 - On	1 - Off
Q3 - PNP	33/13	0 - On	1 - Off	1 - Off
Q2 - NPN	35/19	0 - Off	1 - On	1 - On
Q4 - NPN	11/17	1 - On	0 - Off	1 - On

Figure 3: H-Bridge circuit and corresponding GPIO connections to Raspberry Pi

Figure 4: Added Suspension

Standards:
- IEEE 802.11 (Wi-Fi) Standards
- IEEE 1275 - Open Firmware Standard for Embedded Systems
- Radio Frequency (RF) Standards
- Electromagnetic Compatibility (EMC) Standards

Technical Details
- C++ code base using LiDAR SDK
- Autonomous code handled through Raspberry Pi 4 device powered by external battery pack
- Code was hosted on Git repository shared by team for versioning

Testing
- Two major waves of testing, one in Senior Design Lab and one with physical track
- Design Lab iterative testing focused on determining ranges of LiDAR values for proper turns
- Track testing performed on actual race track focused on reacting to a realistic environment

Final Design

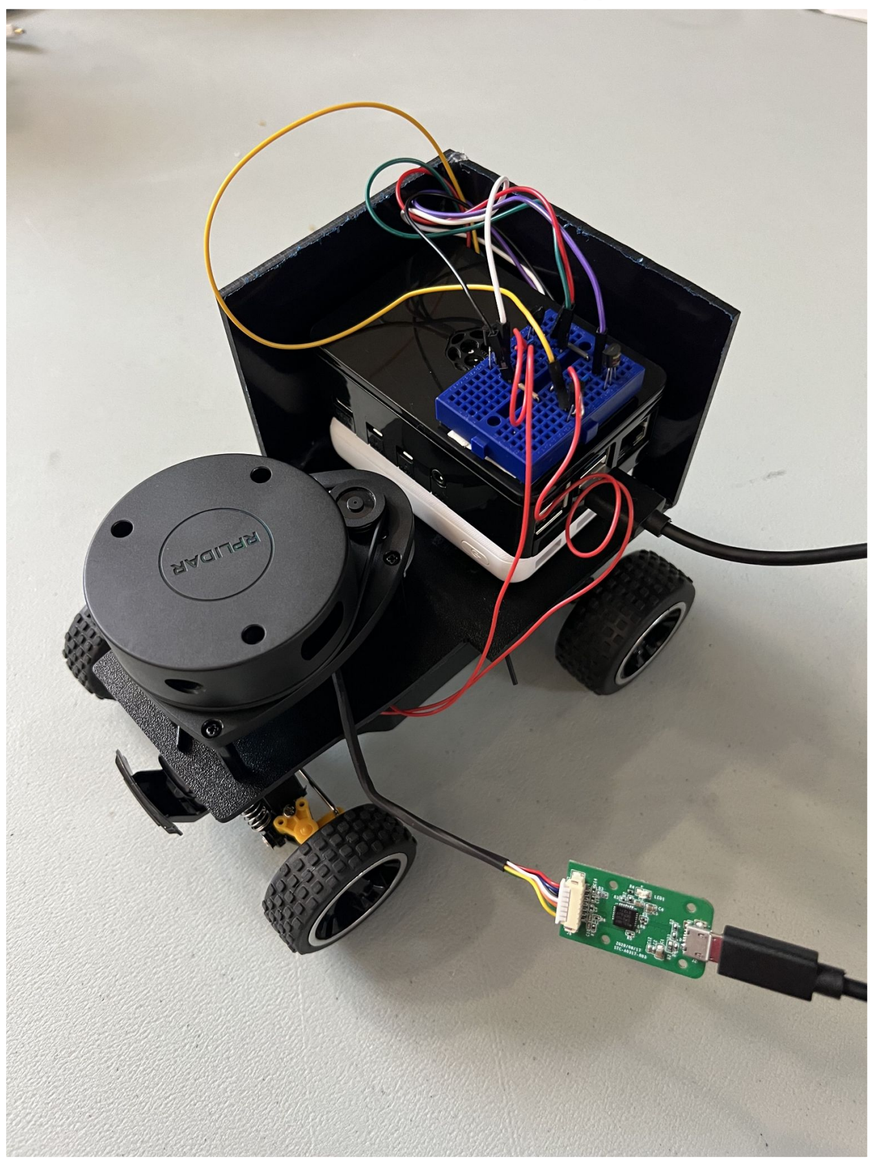


Figure 5: The Complete RC Car