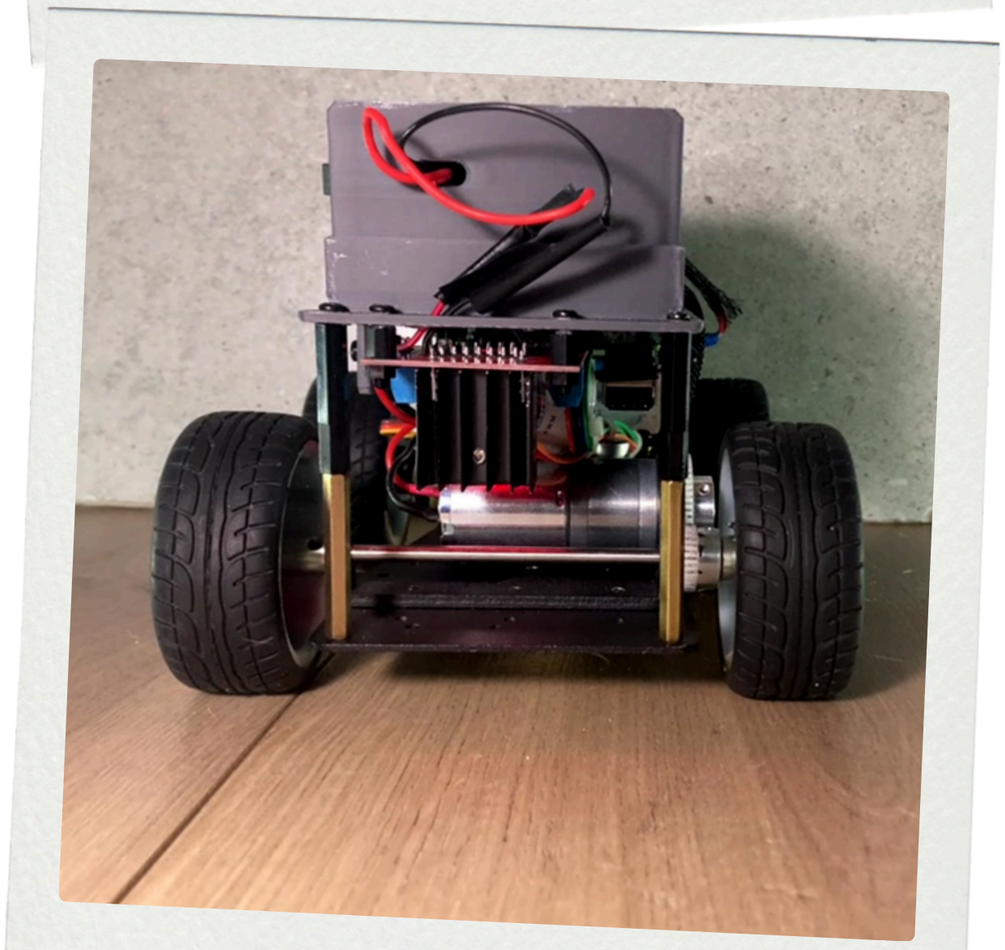
DOCUMENTATION

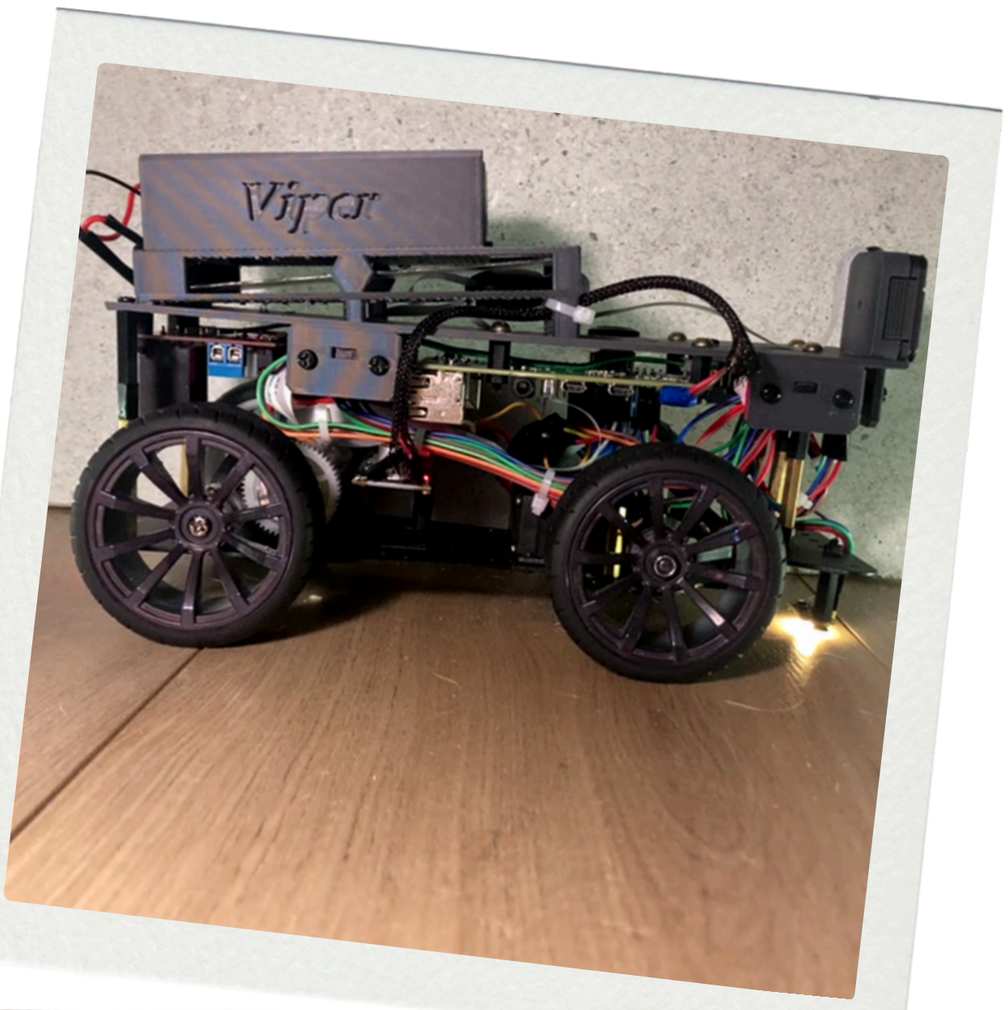
VIPER
WRO FE 2025

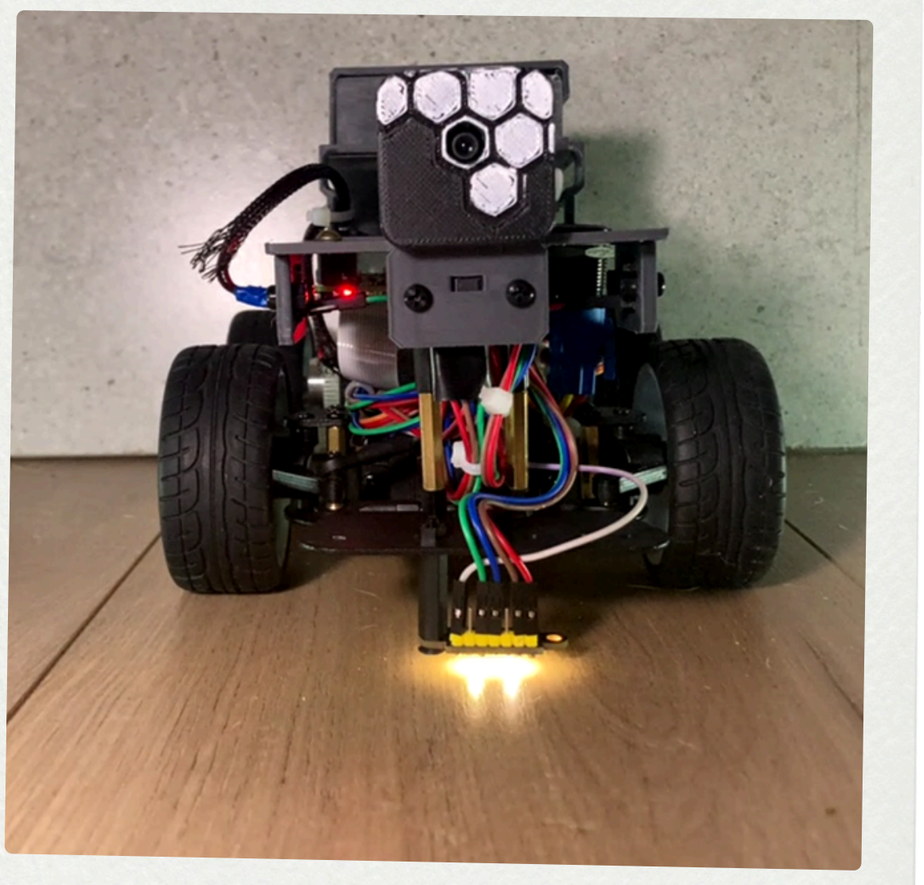
Team Viper

We’re team Viper, a first year Future Engineers team learning, building and adapting as we go. Our name originates from the Python code that powers our robot and the viper snake, known for its precision and adaptability. We try to reflect that in our design: thoughtful, responsive and always improving. Both of us brings different strengths, from coding and mechanical layout to documentation and design. We’re still figuring things out, but we’re proud of what we’ve built—and excited to keep learning.

Team Viper







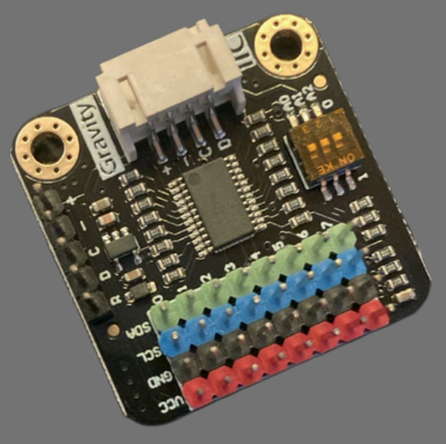
Power and Sense

Our robot operates on a 12V battery pack that serves as the primary power source for both motor control and logic systems. To safely power the Raspberry Pi and its connected peripherals, we use a buck converter to step down the 12V input to a stable 5V output. This regulated voltage is delivered directly to the Pi through a custom USB-C cable, which ensures clean, uninterrupted power tailored to the Pi's requirements. From there, the Pi distributes power to a servo motor, a time-of-flight sensor array via a multiplexer, and a color sensor. This setup maintains electrical isolation between high-current motor operations and sensitive logic components, protecting against voltage spikes while enabling consistent, expressive performance across the system.

For environmental sensing, our robot uses five VL53L0X time-of-flight sensors arranged to provide directional awareness. One sensor faces forward, while two are mounted on the left side—one at the front and one at the rear—and another two are positioned similarly on the right. This configuration helps us detect obstacles and measure distances from multiple angles, improving navigation and control. For color detection, we use a TCS34725 color sensor, mounted on the front underside of the bottom chassis. It enables surface recognition and line tracking by detecting RGB values and ambient light. To manage all these connections, we use a DFRobot I²C multiplexer, which allows multiple I²C devices to communicate with the Raspberry Pi efficiently, avoiding address conflicts and simplifying wiring.

VL53L0X

TCS34725



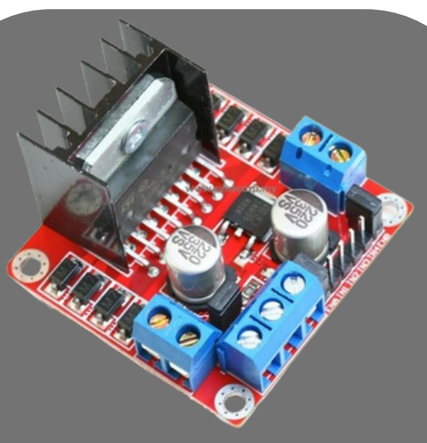
DFROBOT I2C MULTIPLEXER

Visualising

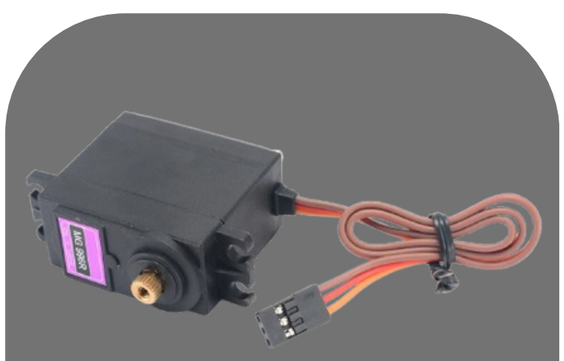
For obstacle management, we use a Raspberry Pi Camera Module 3, mounted at the front of the robot. This module provides high-resolution imaging and fast frame rates, enabling real-time visual detection of obstacles. It interfaces with our onboard Raspberry Pi, which processes the camera feed using custom scripts to identify objects and adjust the robot's path accordingly. The camera's compact form factor and wide field of view make it ideal for navigating tight spaces and dynamic environments.

Mobility

For our drivetrain, we use an MG996R servo motor to control the steering mechanism at the front of the robot. It provides reliable torque and precision for directional changes. Our main propulsion comes from a 12V DC motor connected to a gearbox, which helps regulate speed and increase torque for smoother movement. To manage motor control, we use an L298N dual H-bridge motor controller, which allows us to drive the DC motor forward and backward with PWM speed control. We've also shortened the wheelbase from the original model we were provided, which improves maneuverability and gives us more flexibility in chassis layout. These adjustments help us balance control, power, and space efficiency as we continue refining our design.



L298n



MG996R

Pseudo Codes

Round 1

```
Start the robot
Initialize sensors and motors

Begin forward motion at slow speed
Use side sensors to maintain straight trajectory

Loop:
   Continuously check ground color beneath robot
   If a color is detected:
      Stop and identify the color
      If color is blue:
         Set direction to counterclockwise
      If color is red:
         Set direction to clockwise
      Track color appearances
      If same color appears 4 times:
         Increment lap counter

Use forward-facing sensors to scan ahead
If obstacle detected:
   Check left and right for available space
   Turn toward the side with more space

If path is clear:
   Use side sensors to stay straight
   Adjust steering to remain centered

If lap counter reaches 3:
   Break loop

Execute parallel parking maneuver
Stop the robot
```

Round 2

```
Start the robot
Set up sensors, motors, and camera module

Move forward slowly
Use side sensors to stay straight

Loop:
   Check ground color under the robot
   If a color is detected:
      Stop and identify the color
      If color is blue:
         Set direction to counterclockwise
      If color is red:
         Set direction to clockwise
      Track color appearances
      If same color appears 4 times:
         Increment lap counter

Use camera to look ahead
If a block is detected in front:
   Identify block color using camera
   If block is red:
      Turn right to go around the block
      Resume forward motion
   If block is green:
      Turn left to go around the block
      Resume forward motion

If no block detected:
   Check for obstacles using proximity sensors
   If obstacle detected:
      Check left and right for space
      Turn toward side with more space

   If path is clear:
      Use side sensors to stay straight
      Adjust steering to stay centered

   If lap counter reaches 3:
      Break loop

Parallel park
Stop the robot
```

Parts (Electronic):

Part:	Amount	Website
VL53L0X	5x	Link
DFROBOT I2C Multilpexer	1x	Link
Raspberry PI	1x	
32 GB Micro-SD	1x	
Raspberry Pi camera module 3	1x	
Ribbon cable *	1x	Link
TCS34725	1x	Link
USB-C	4x	Link
10cm cable pack (60)	2x	Link
20 cm cable pack (60)	1x	Link
Motor controller	1x	Link
DC motor	1x	
Battery	6x	Link
3x Battery sled	1x	Link
Buck converter	1x	Link
16 mm button momentary	1x	Link
Rocker switch	1x	Link

Parts (Chassis):

Part:	Amount	Website
Upper Chassis (STL)	1x	Link
Lower Chassis	1x	
Spacer kit	1x	Link
Wheels	4x	
Steering Mechanism	1x	

STL Model	Website
VL53L0X Mount	Link
Camera module 3 mount (Back)	Link
Camera module 3 (Front)	Link