RoboNYC

Jessica Wang | Audrey Shaker | Griffin Genieser

Trinity School NYC

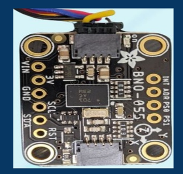
2025 RoboCup Junior Internationals

Scan the QR code for more RoboNYC information!

ABSTRACT: Our goal was to find the most reliable and efficient way for our robot to function. We went through many iterations for PCB, CAD, and finding the best sensors. We use a double-offense strategy. Please use the QR code in the top right corner to see full details.

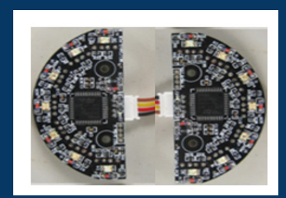
SENSORS

ADAFRUIT BNO055



The first IMU we used, the UM7, produced values that fluctuated rapidly, increased every 0.1 seconds without changing its orientation, and needed lots of time to calibrate. As a result, we changed our IMU to the BNO055, which did not take much time to calibrate and produced much more reliable and accurate orientation.

JOINMAX IR SENSORS

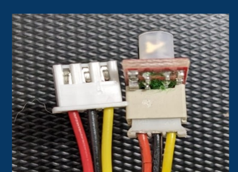


We picked these IR sensors because their fan shape makes them easy to mount and testing revealed that their IR intensity readings are accurate, allowing us to calculate more precise angles for the robots' movement.

DFROBOT URM09 ULTRASONIC SENSORS

We decided to use these ultrasonic sensors because they use I2C protocol so we can stack multiple devices on one Teensy port, and are compatible with our Teensy's pinout layout with the rest of our components.

ZOBOT GRAYSCALE SENSORS



We decided on these analog grayscale sensors to detect the white boundary lines after researching other teams' line detection sensors.

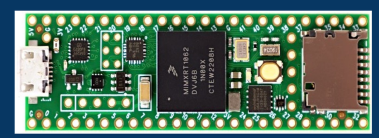
DFROBOT ANALOG FLAME SENSOR V2

We chose this flame sensor to detect whether the ball is in the ball capture zone because we found through our research that it was reliable and accurate.

APPLICATIONS USED

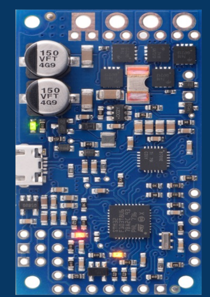
NON-SENSORS

TEENSY 4.1



Last year, our team used the Arduino UNO as the microcontroller for our robot, which took up lots of space and didn't have as many pins. As a result, the previous robot had a limited amount of components that it could fit. Thus, due to the Teensy's compact size, sufficient number of pins, and high processing power, we decided to use the Teensy as our main microcontroller.

SIMPLE MOTOR CONTROLLER G2 18v15

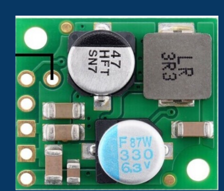


We use this motor driver as it is versatile and very space efficient. This motor driver also allows for custom soldering, letting us customize it to our needs.

OMNI WHEELS

Our wheels are standard aluminum omni-directional wheels. Compared with other omni-directional wheels, they are thin, lightweight, and fast.

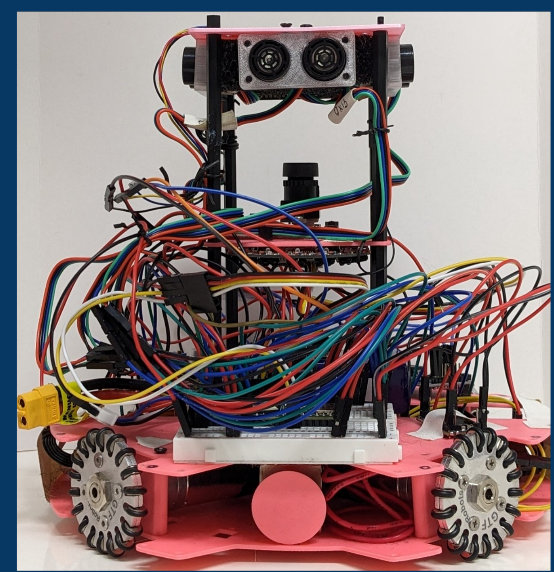
POLOLU 5V STEP-DOWN VOLTAGE REGULATOR



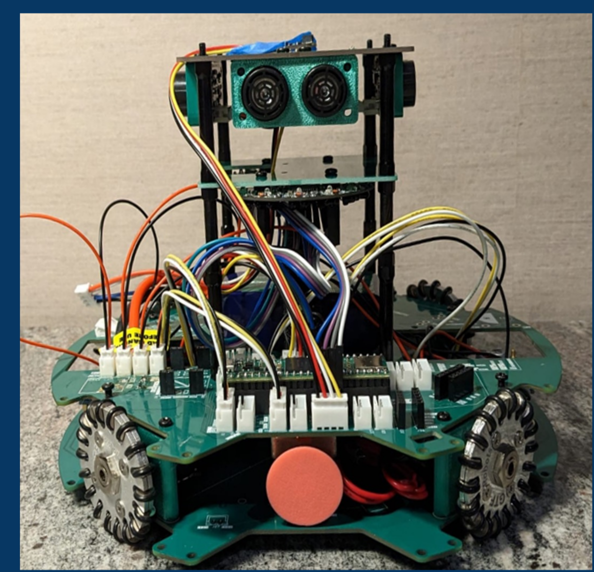
We tested several voltage converters including ones that stepped down to 3.3V, and found that this Pololu model is the the smallest and most reliable.

11.1V TATTU 1800MAH 3S 75C LIPO BATTERY

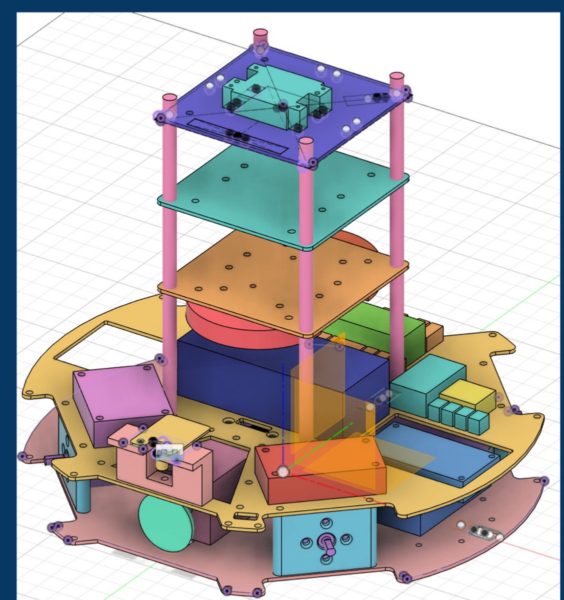
We decided to use this battery as we needed 12V to power our motor drivers, and we chose this battery for its high capacity of 1800mAh so that it would not run out of energy before half time. Originally, we used an 11.1V lipo battery with 1300mAh, and it ran out of energy too quickly.



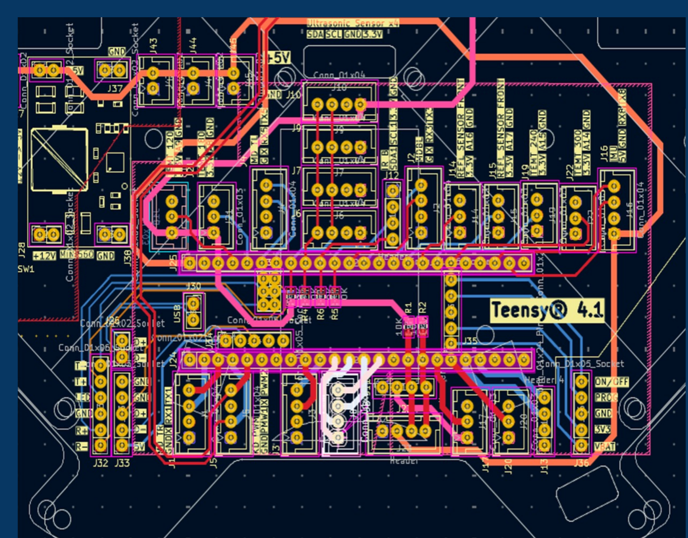
PROTOTYPE WITH BREADBOARD



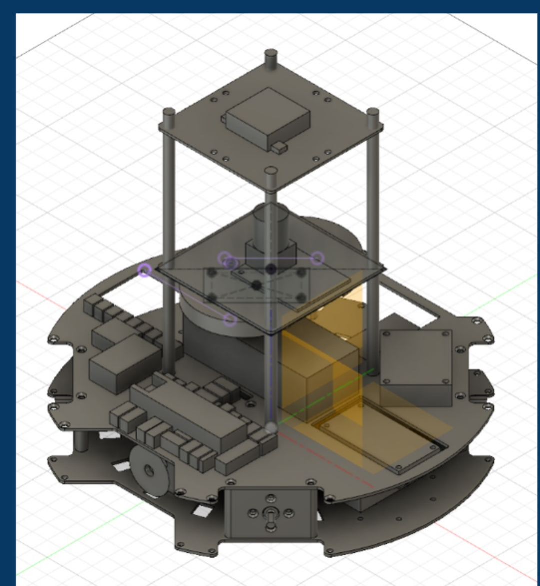
FINAL DESIGN WITH PCBS



3D CAD



PCB DESIGN



EARLIER VERSION OF CAD

ROBONYC AT USA ROBOCUP JUNIOR COMPETITION

SOFTWARE

OVERVIEW

We used Arduino to program our robot. Our original plan was to have a striker and goalie robot. We prioritized coding the striker robot, which shoots goals, before working on the goalie robot, which stays in the penalty area. However, we soon realized the goalie robot was unreliable and sometimes ineffective. Therefore, we made both our robots a more consistent offense, as they are identical.

HOMING

Our homing program allows our robot to move to a fixed home position in front of our goal when it cannot see the ball. The robot also homes for one second when the grayscale sensors detect the white lines, indicating that the robot has reached the field boundaries.

MOVEMENT

If the ball is almost directly in front of the robots, they chase it at the angle they see it. If not, the robots circle around so they are always facing the opponent's goal. To achieve this, we used vector trigonometry to calculate the speeds of each wheel at a given angle.

AIMING

When the flame sensor detects that the ball is in the ball capture zone, the robot will then check to see whether it is in the opposing side of the field using its front and back ultrasonic sensors. If it determines that it is on the opposing side of the field, it will execute scoring mode. In scoring mode, the robot checks again to see whether it is on the left or right side of the field. Depending on the result of the condition, the robot will then proceed to drive and turn at the same time either leftward or rightward to attempt a goal.

SENSOR BOUNDARIES

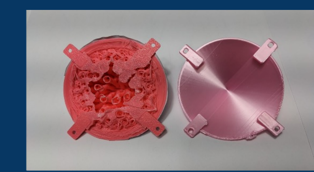
Ultrasonic sensors provide drive boundaries for the robot through vector trigonometry, ensuring the robot is always at a safe distance from the boundaries of the field.

ITERATIONS

BREADBOARD

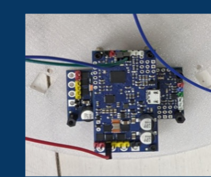
Our first complete robot operated with a breadboard, which served as its entire electrical design. The many wires involved in this approach often became tangled or interfered with our IR sensor readings. We decided to design PCBs to use as our robot's plates so the electrical connections would be more reliable.

CAD



Our CAD design also went through many changes, with our original design including a dribbler, kicker, and camera with a hollow conical mirror. We decided to leave out these components due to size and weight limitations, but we hope to implement them in the future. We especially spent a lot of time on the CAD for the mirror so that it would preserve its hollow shape.

MOTOR DRIVERS



We deliberated the placement of our motor drivers, which we first mounted in a criss-cross design between the 1st and 2nd plate, then moved it to the top for ease of access and protection from collisions.

JUMPING WIRES ON PCB

When the PCBs arrived, we found out that 3.3V from the Teensy was not enough to power all of our sensors, especially in the cases where multiple sensors were routed to the same Teensy pin. We remembered that the Teensy had a 5V port that was equally capable of powering the sensors, so we jumped cables underneath our PCB to connect the 5V power supply to the first pin with a sensor that was supposed to receive 3.3V, which was internally routed in the PCB to all of the other 3.3V sensors. Now, with a greater voltage supplied, the sensors were able to perform much better.