Method, Production & Design

Hardware Overview

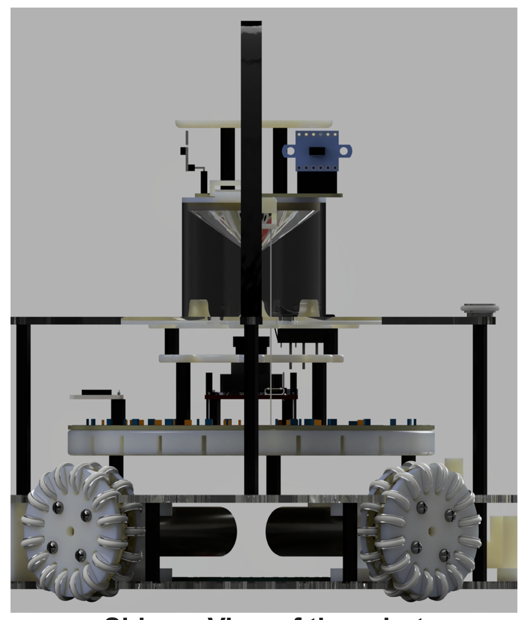
The structural design of the robot was created in Autodesk Fusion 360, and the PCBs were designed in Autodesk EAGLE. These programs were used because the file types were compatible, making it easier to visualise a finished robot and simulate the fit of components. Custom-made parts, incorporated in the design, were either laser cut or 3D printed. We designed the PCBs and ordered from a fabrication manufacturer. We have worked for over 500 hours and spent approximately AUD2500 on parts for the current robot.

Chassis design

The robot is comprised of three separate layers that form the frame of the robot. This creates a robot that is easily accessible for both production, maintenance and repairs. The chassis is made up of three laser cut polycarbonate plates:

- The bottom plate is designed to hold four DCX19 Maxon motors, 32 LEDs and phototransistors (light sensors) and a battery holder.
- The middle plate supports the main PCB and provides additional support to the motors.
- The top plate is used to stop the robot from entering the goal area and protects the electrical components from accidental impacts.

The three plates are connected by lightweight nylon standoffs.



Side on View of the robot

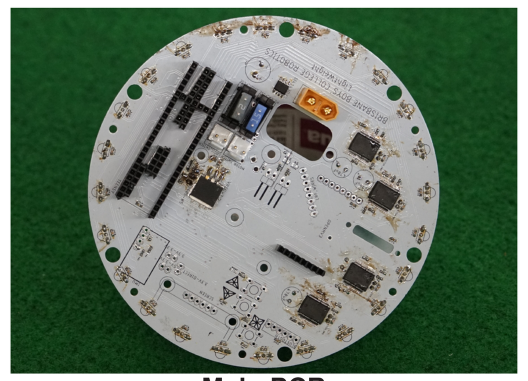
Electrical components

The robots' electrical system is a combination of three PCBs. PCBs were used to reduce the number of external connections, increasing the reliability of the connection to different components. The PCBs are connected to each other by flat flex connectors (FFCs) similar to FFCs used in many smart-phones and tablets. The PCB on the bottom plate (light sensor PCB) is used to detect the white line on the field. The light sensor PCB holds:

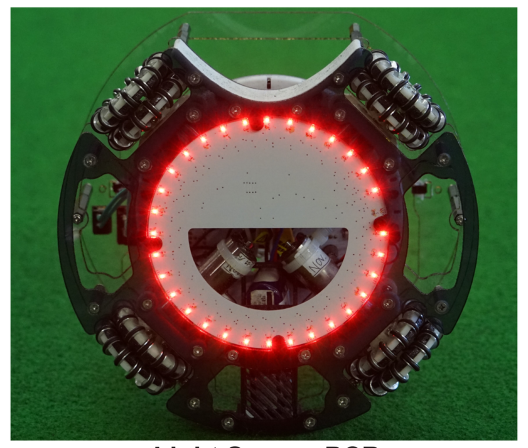
- 5-bit multiplexer
- 32 LEDs
- 32 phototransistors.

The PCB on the middle plate (main PCB) holds all the main circuitry and the bulk of the components. It holds the two micro-controllers that calculate the robots movement. The main PCB consists of:

- 1 Teensy 3.5 (micro-controller)
- 1 OpenMV H7 camera (micro-controller)
- 24 TSSPs (infrared sensors)
- 4 VNH5050 (motor controllers)
- 1 HC-05 (bluetooth module)
- 3.3V and 5V power supply for the whole robot.



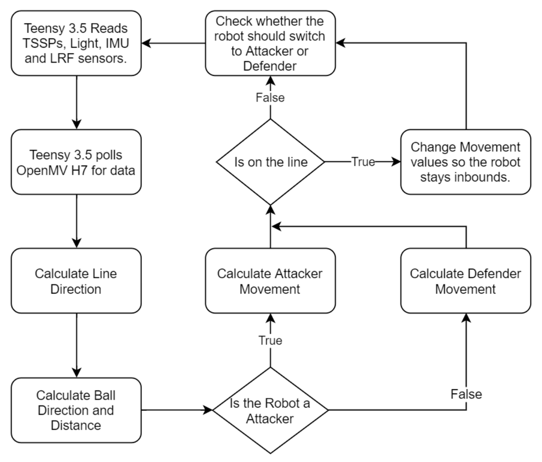
Main PCB



Light Sensor PCB

Software Overview

The robots have been programmed using C++ and Python with two micro-controllers on each robot. The Teensy 3.5 does the bulk of the calculations and receives data from all the sensors. The OpenMV H7 is an optical sensor that calculates the goal direction and distance and sends those values to the Teensy. The Teensy then sends motor values to the motor controllers. The robot follows a set of steps, shown in the Robot Logic Flowchart, to determine where it should move.



Robot Logic Flowchart

Communication & game strategies

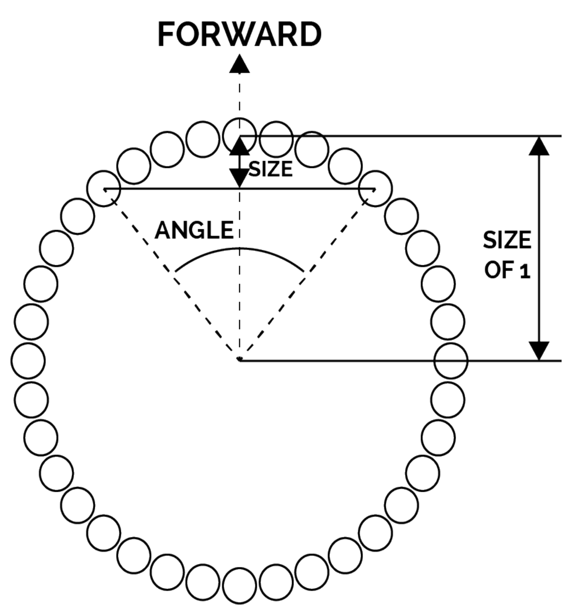
Each robot has an HC-05 bluetooth module that pair with each other. The robots can send their position (attacker or defender); the ball direction; and the ball strength (distance) using a serial protocol. The robots can decide whether they should be an attacker or a defender and switch positions accordingly, just like real soccer players. This strategy is implemented in gameplay and helps to reduce the number of double defence fouls.

Line avoidance

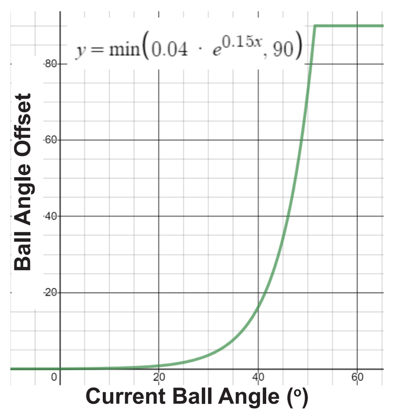
The robots incorporate 32 light sensors to detect the white line and stay in the field. The light sensors are arranged in a ring, making it easier to perform vector calculations. The angle of the line is calculated using vector addition. This data is sent to the Teensy 3.5 and the robot moves in the opposite direction to the line, so it stays in bounds.

Ball following

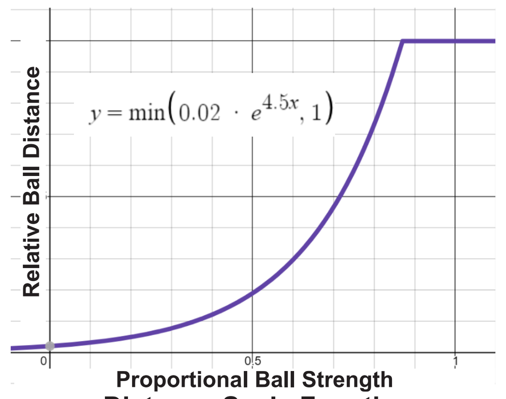
Each robot has 24 digital IR sensors with an output value of 0 when it detects IR light, or an output value of 1 when it can't detect IR light. The direction of the ball is calculated using vector addition. After the direction of the ball is calculated, two mathematical models calculate where the robot should move. The first model determines the angle added to the ball direction, so the robot positions itself behind the ball (orbit function); the second model scales down the angle added based on how far away the ball is (distance scale function).



Line Tracking



Orbit Function



Distance Scale Function

Abstract

M&A is an Australian team from Brisbane Boys' College. Our team of four have made two identical robots to compete. Last year we placed 2nd at the Australian National Competition. The robots have been developed since September 2017 with five different design iterations for different RoboCup Competitions.

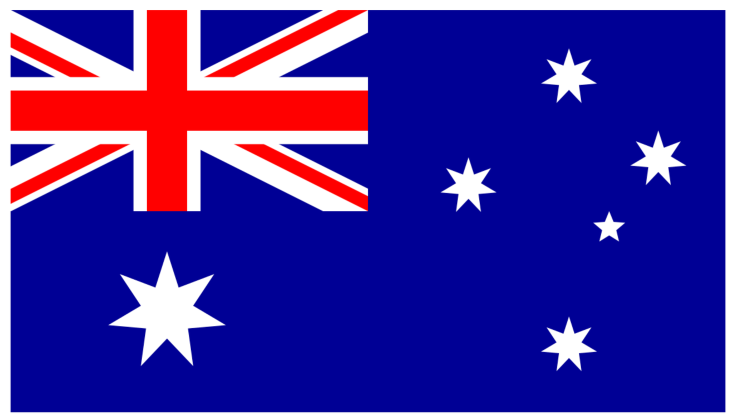
The robots were built using industry standard technologies - 3D printing, laser cutting, vacuum forming - and incorporate printed circuit boards (PCBs) to increase efficiency and reliability. Our robots also include laser range finders to track the robots' position on the field. The robots communicate via bluetooth modules to optimise gameplay performance.

This poster outlines our findings on how to best build a robot for this competition to share our knowledge with other teams. We want to help RoboCup achieve their goal of making robots that can beat the best soccer players in the world by 2050.

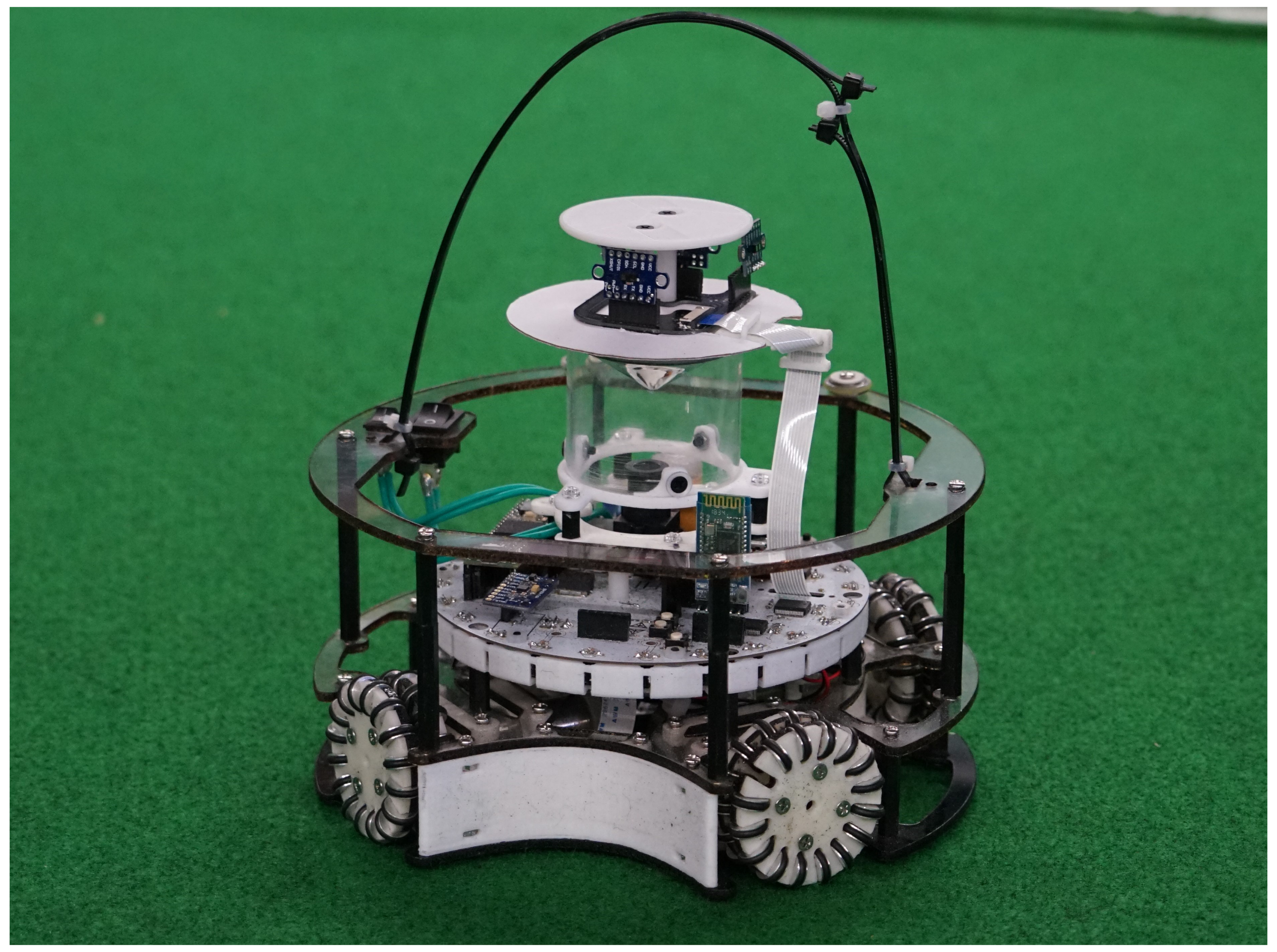
Team M&A

M&A

BRISBANE BOYS' COLLEGE - JUNIOR LIGHTWEIGHT SOCCER



ROBOTICS
BRISBANE BOYS' COLLEGE



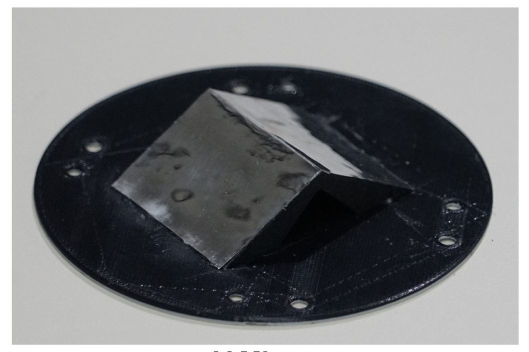
M&A's Robot

Data, Results & Discussion

Testing & Development

During the production process for the robot, there were many challenges to overcome. The best solution was found by testing different methods and comparing the data and results from those tests. The three main components that were developed were the camera, wheels and the laser range finders (LRFs).

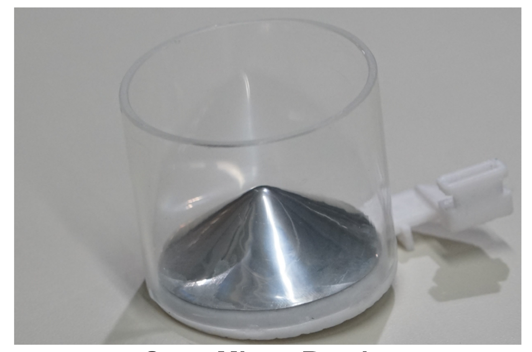
Camera



V Mirror

Pros:
- View distance is increased
- Easy to manufacture

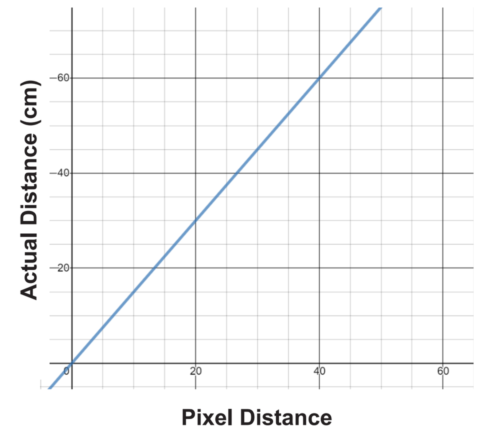
Cons:
- Cannot see sideways
- Image distortion
- **Worse for gameplay**

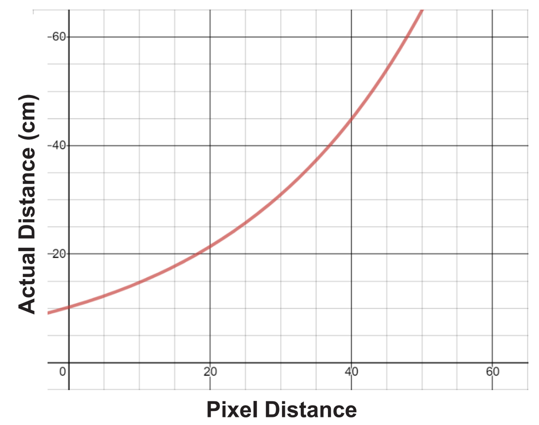


Cone Mirror Render

Pros:
- Can see in all directions
- Reduced image distortion
- **Better for game-play**

Cons:
- View distance is reduced
- Harder to manufacture





Graphs of the relationship between the pixel distance and the actual distance for both mirrors

The 360-degree camera vision system is comprised of the OpenMV H7, a custom-moulded mirror and acrylic tube. The camera is positioned so that it looks vertically at the mirror, seeing the entire field. There were two main stages in the development of the mirror shape. The first mirror shape is a V while the second is a cone. A series of brackets allow the cone to be positioned to the centre of the camera's field of view. A circular piece of card is placed above the cone to protect the camera from glare, which would wash out the image.

To confirm the relationship between the distance in pixel units to the distance in centimetres, multiple tests were undertaken. The results of these tests produce the models (to the left) which can be implemented in the software logic to calculate the robots' position on the field.

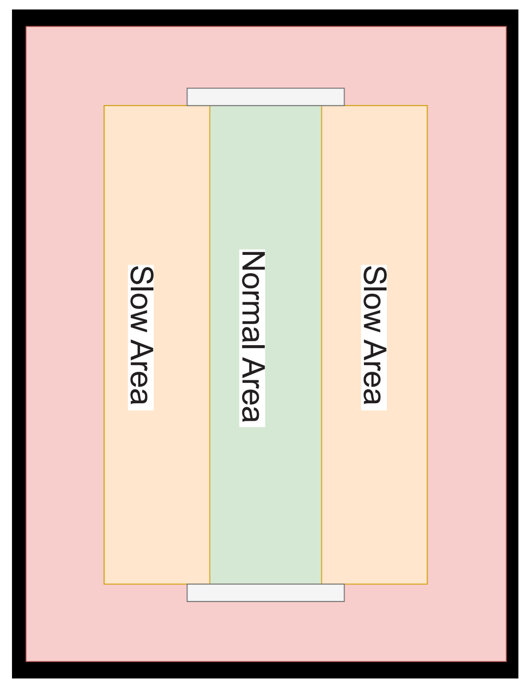
The mirrors on the robot were made using a vacuum-forming process. The PVC mirror material was first heated, then placed on a mould, where the material was vacuum-formed over an impression to create a detailed mould. This made it easier for the robot to position itself.

Wheels

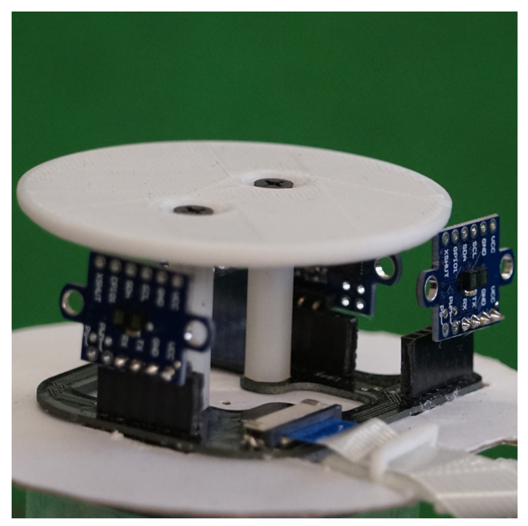
An early iteration of this robot included the pre-made omni-wheels from GTF-Robotics. These were an easy to use omni-wheel; however, they lacked in grip which caused the robot to slide as it attempted to avoid the line. The rubber o-rings on the wheels wore down very quickly and we couldn't find a suitable replacement. A new wheel was designed to fix this issue. It included larger rollers that were produced by a CNC lathe to have more grip on the floor, and the current iteration of this includes two wheel layers to maximise contact with the floor.

Wheel Progression - GTF Wheel, Single Layered Custom Wheel, Double Layered Custom Wheel (Left to Right)

Laser Range Finders



LRFs Avoidance Coordinates



LRF PCB

LRFs are distance sensors that calculate the distance between itself and the nearest object. These sensors are used in collaboration with the camera to calculate the robots' position. Due to gameplay constraints, there was difficulty in finding a suitable LRF sensor that can accurately detect objects in the field. After testing multiple LRF sensors, the VL53L0X (GY-53 breakout) LRF sensor was deemed the most suitable sensor for gameplay.

The Software implemented on the robot uses a fusion of LRF and Light Sensor values to stay in bounds during gameplay. The LRFs can detect when the robot enters the edge of the playing field and slows down in this area. This is so the robot has a lower chance of going over the line. The LRFs send data to the Teensy 3.5 via a Serial protocol. It sends 16-bit values and has millimetre accuracy.

For more information visit our website via this QR code