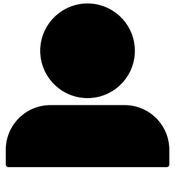
BOT STRIKE



Presented by
Zaid Balout
Khaled Yaish

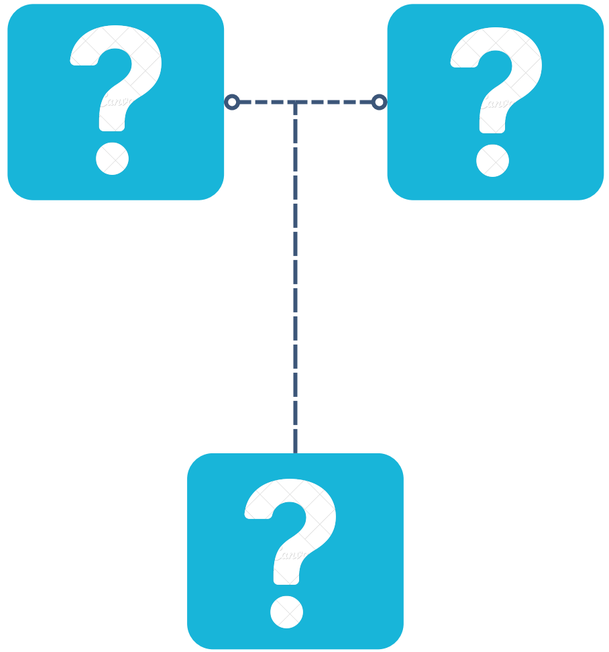
Supervised by
Dr. Anas Toma

Inspiration

Crash Bash Game

BotStrike is inspired by classic arena games like **Crash Bash**, where players control characters inside a closed arena and compete in fast-paced matches.

What is BotStrike?

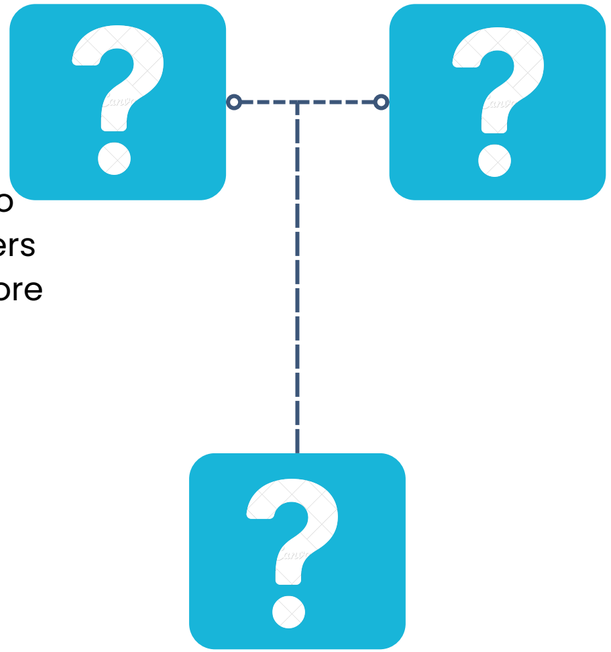


How does BotStrike work?

What makes BotStrike unique?

What is BotStrike?

BotStrike is a tabletop arcade game with up to **four robots** moving on a **hexagon** arena. Players use **Controllers** to control robots and try to score goals.



How does BotStrike work?

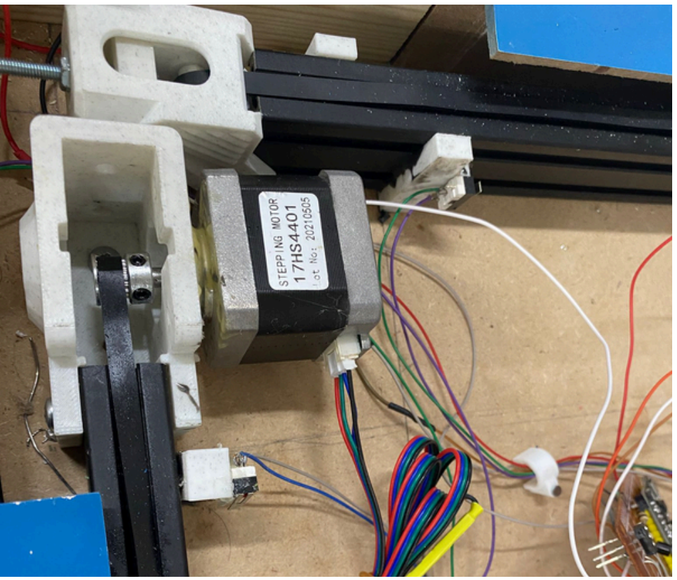
- **Control loop:** Joystick → Microcontroller → Motor driver → Robot moves.
- **Vision loop:** Camera → Laptop processing (crop + mask color) → Ball coordinates → AI decision → Commands to robot.
- **Game logic:** Goals detected by Laser + LDR, Arduino handles score & sounds, reset on win.

What makes BotStrike unique?

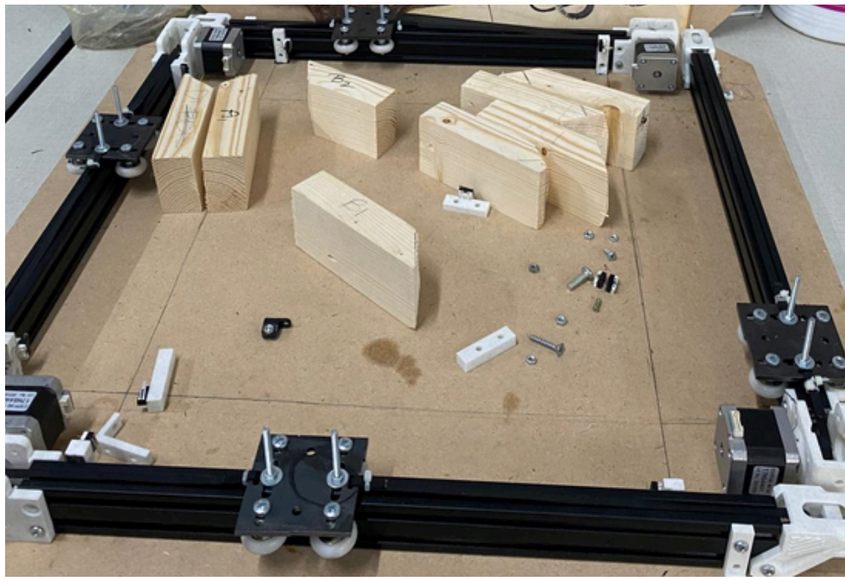
- **AI** as a replacement player: the camera watches the field and the AI **controls** robots when a player are missing.
- Real hardware + real-time control: custom arena, robots, and electronics.
- Stable, responsive gameplay proven in testing (low latency control, reliable power, smooth motion).

Components Used in BotStrike

Arena (Game Board)

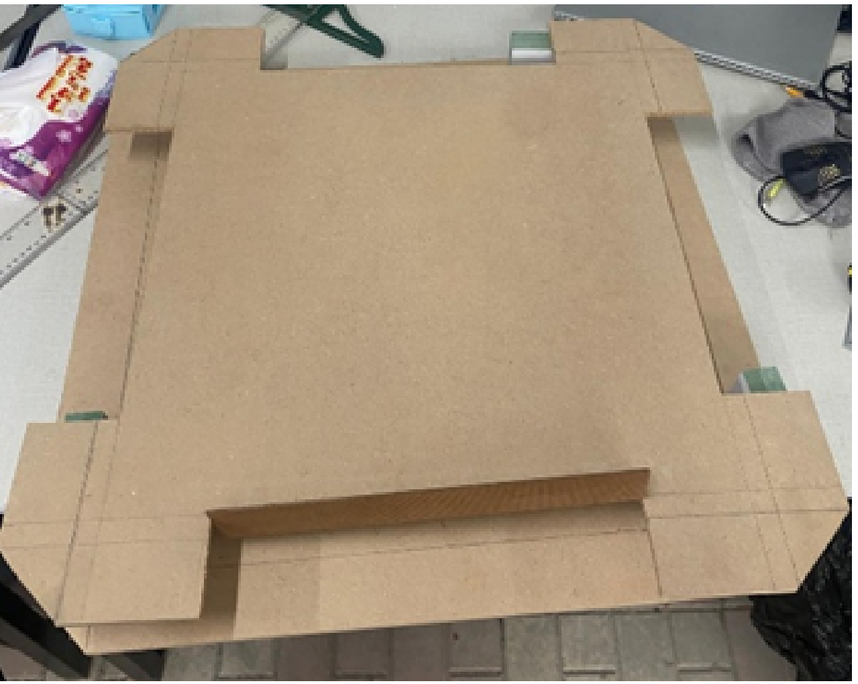


- The arena is a hexagon-shaped board made from MDF and aluminum profiles for strength and stability.
- Each side has a goal area and a place for a robot.

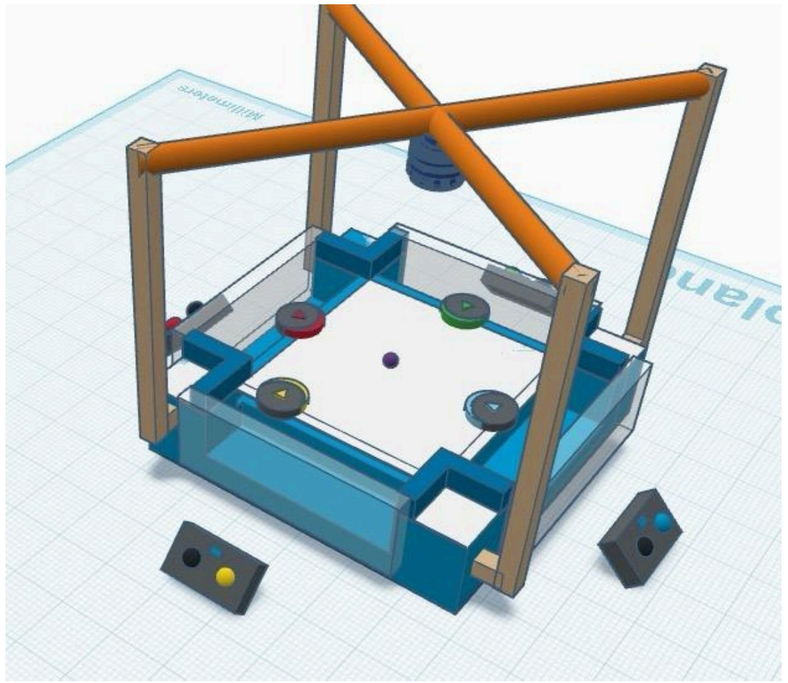


Components Used in BotStrike

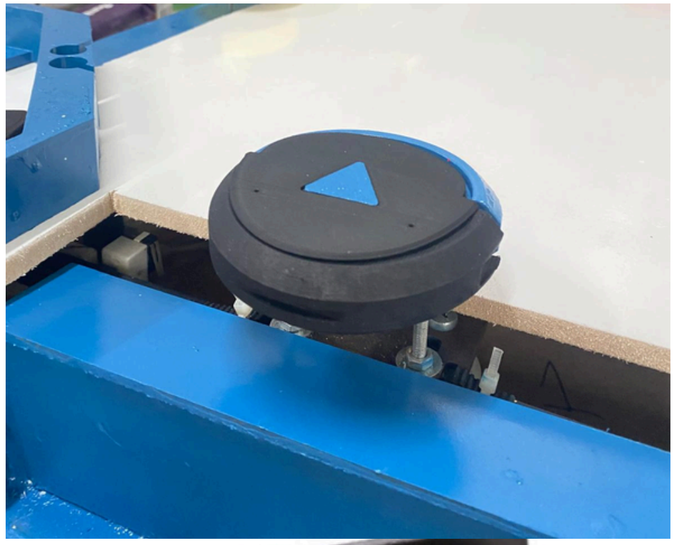
Arena (Game Board)



- On the top, there is a camera that watches the game.
- The design makes the game fair, fast, and fun for both players and AI robot.



Components Used in BotStrike

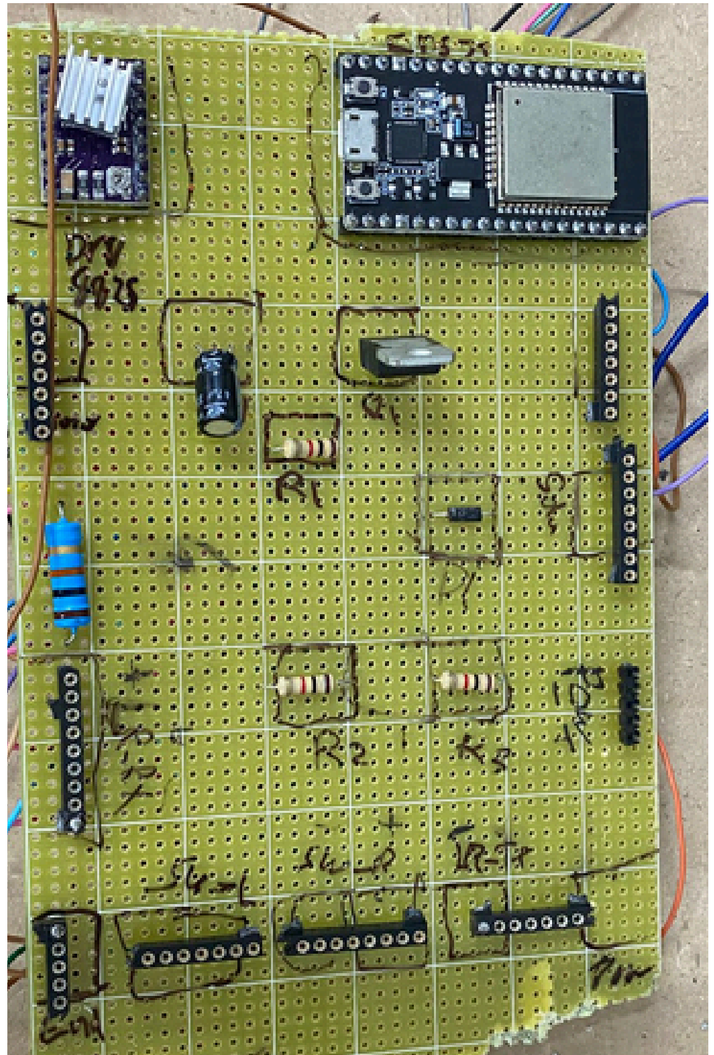


Player Robots

- Each robot is built on a metal square frame that slides on aluminum profiles.
- The frame has 4 wheels for smooth and stable movement along the tracks.
- Two screws are used to mount the external plastic body, which is the colored part players see.
- Designed to be lightweight but strong, so it can handle fast gameplay without breaking.
- Controlled by motors and drivers, allowing quick direction changes.

Components Used in BotStrike

Motor Control Boards (x4)



- One board per robot to keep wiring short and enable real-time control.
- ESP32 on each board handles local control and communication.
- Buck converter regulates the main supply to 12V / 5V to protect components.
- 8855 stepper driver receives DIR/STEP/EN (timed) and drives the stepper (4 wires) along the aluminum profile.
- Solenoid output is switched by a transistor with a flyback diode for safe actuation.
- Headers/connectors for motor, sensors, and power simplify maintenance and replacement.
- Clean routing + stable power for continuous play (reduced noise and fewer glitches).

Components Used in BotStrike

Controllers (Joysticks)

Each player holds a 2-axis analog joystick. The signals go directly to the microcontroller for low-latency and responsive control, so the robot movement feels instant and natural during play.

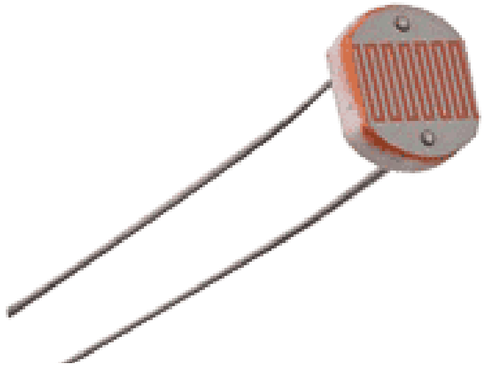
Components Used in BotStrike

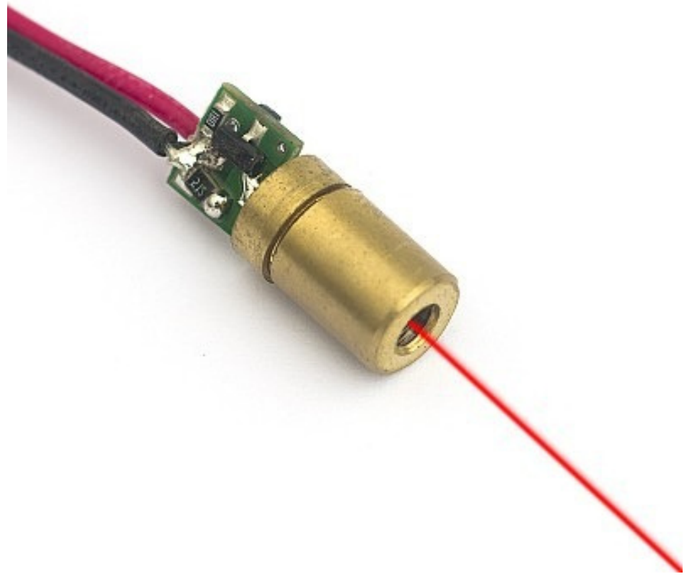
Limit Switches (End-stops)

Each robot has two limit switches. They are wired active-low (3.3 V becomes 0 V when pressed). These switches act as safety stops so the carriage does not run off the aluminum track, and they help keep the motion accurate during fast matches.

Components Used in BotStrike

Goal Sensor (Laser + LDR)





We detect a goal with a 5V laser facing an LDR on the other side of the goal.
The ESP32 reads the LDR with the ADC, when the laser beam is broken, the analog value drops below a threshold, and we mark a goal event immediately. This method is simple, fast, and reliable during matches.

Components Used in BotStrike

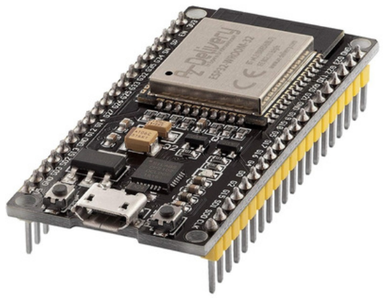
Solenoid Kicker

To hit the ball, the robot uses a 12V solenoid controlled by a transistor with a protection diode. When the player presses the kick button, the ESP32 sends a short pulse to the solenoid. This creates a quick push that kicks the ball without consuming too much power.

Components Used in BotStrike

Networking

Server



ESP 32

Client

Laptop

This pairing is easy and stable: the client searches by device ID, connects to the ESP32, and then sends movement commands based on AI decisions.
Events like goals also travel back for scoring and audio.

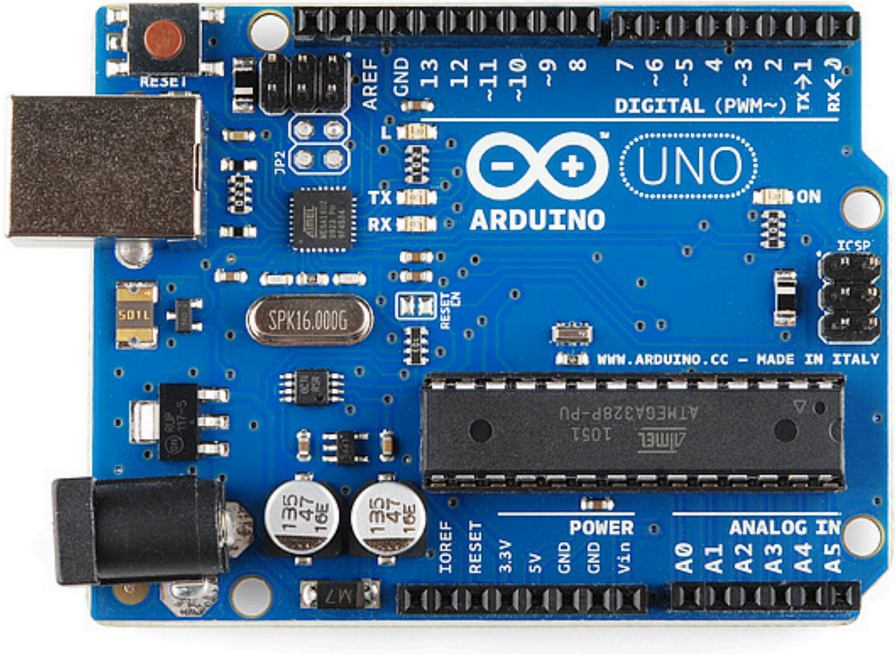
Components Used in BotStrike

Camera & AI

The camera stream goes to the laptop.
We crop the image to the arena area and apply a pink color mask to isolate the ball.
When the pink ball is found, we compute its (x, y) and map that to board coordinates. If the ball is to the left or right, we command the robot to move
if the ball is close, we trigger the solenoid push. We tuned thresholds for light conditions to keep detection stable.

Components Used in BotStrike

Arduino Audio & Side Buttons



For the game feel, the Arduino plays a main music loop and a goal sound when a goal happens. Audio files are on a DFPlayer with an SD card.
The signal goes to an amplifier and then to the speaker. On the side panel, we have NEXT, PREVIOUS, VOL UP, VOL DOWN, PLAY/PAUSE buttons to control audio during the match.

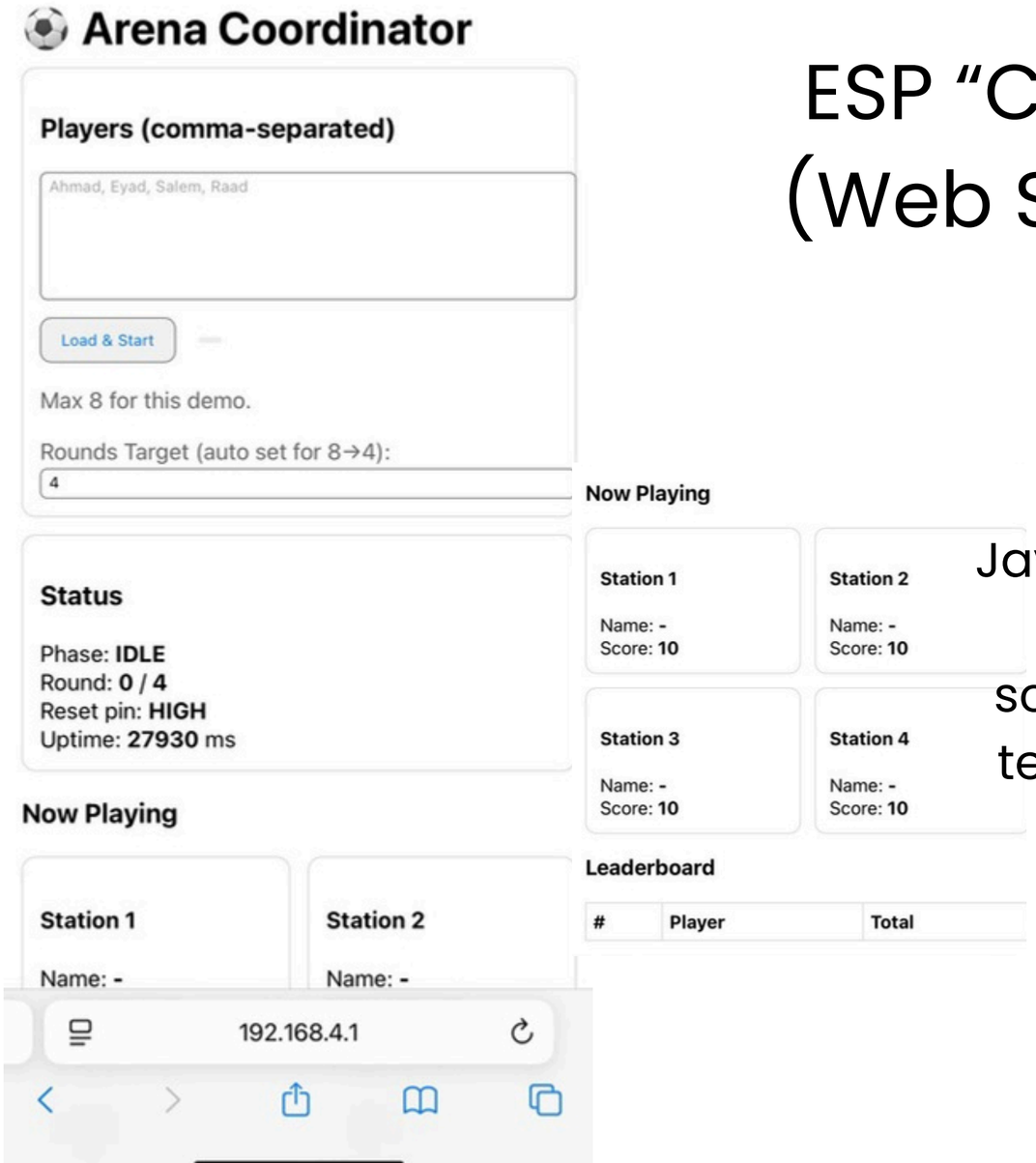
Components Used in BotStrike

Scoring Logic (FFA & 2v2)

The Arduino tracks lives/score for each robot.
In Free-for-All, when three robots reach zero, the last robot wins and we reset.
In 2v2, when both robots on a team reach zero, the other team wins and we reset.
We also mirror goal signals between teammates (e.g., Robot 1 ↔ Robot 2, Robot 3 ↔ Robot 4) so each pair stays in sync.

Components Used in BotStrike

ESP “Coordinator” (Web Scoreboard)



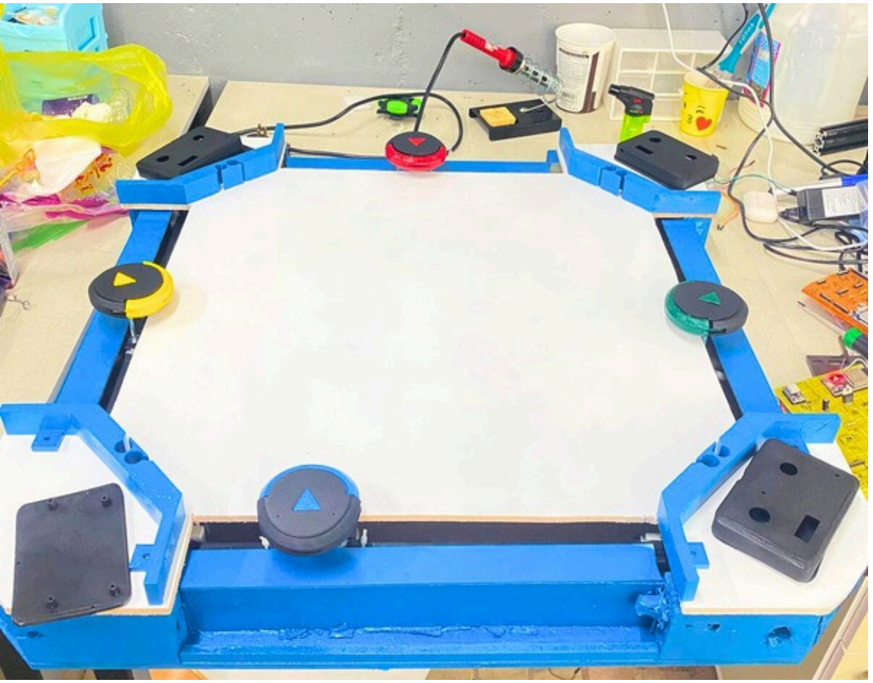
The Coordinator listens to goal events and serves a small Wi-Fi web page (HTML + JavaScript). You can enter player names, create a mini-league, and see live updates of the scores. On every reset, it records the result and tells players which corner (Robot 1–4) to use in the next match, so the tournament runs smoothly.

Components Used in BotStrike

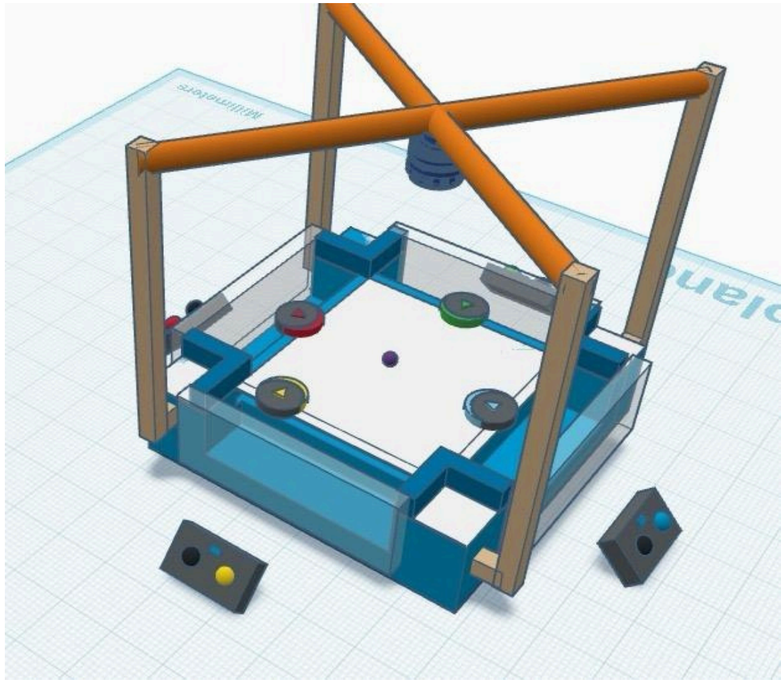
Power & Reliability

We use a stable main supply, a buck converter for logic voltage, and a separate 12V path for the solenoids. Clean wiring and connectors reduce noise and make debugging and replacement fast. This setup kept the system stable during long play sessions.

Final Prototype



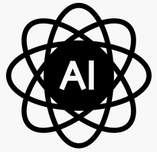
Our final prototype is a fully working tabletop arcade game built from scratch. It includes a hexagon arena, four player robots with joysticks, and an overhead camera with an AI module to control missing players. The build is stable and responsive during play.



Constraints

01 High Budget – The project cost was high because parts were overpriced in our local market.

02 Overhead camera sometimes struggles with tracking under poor lighting.

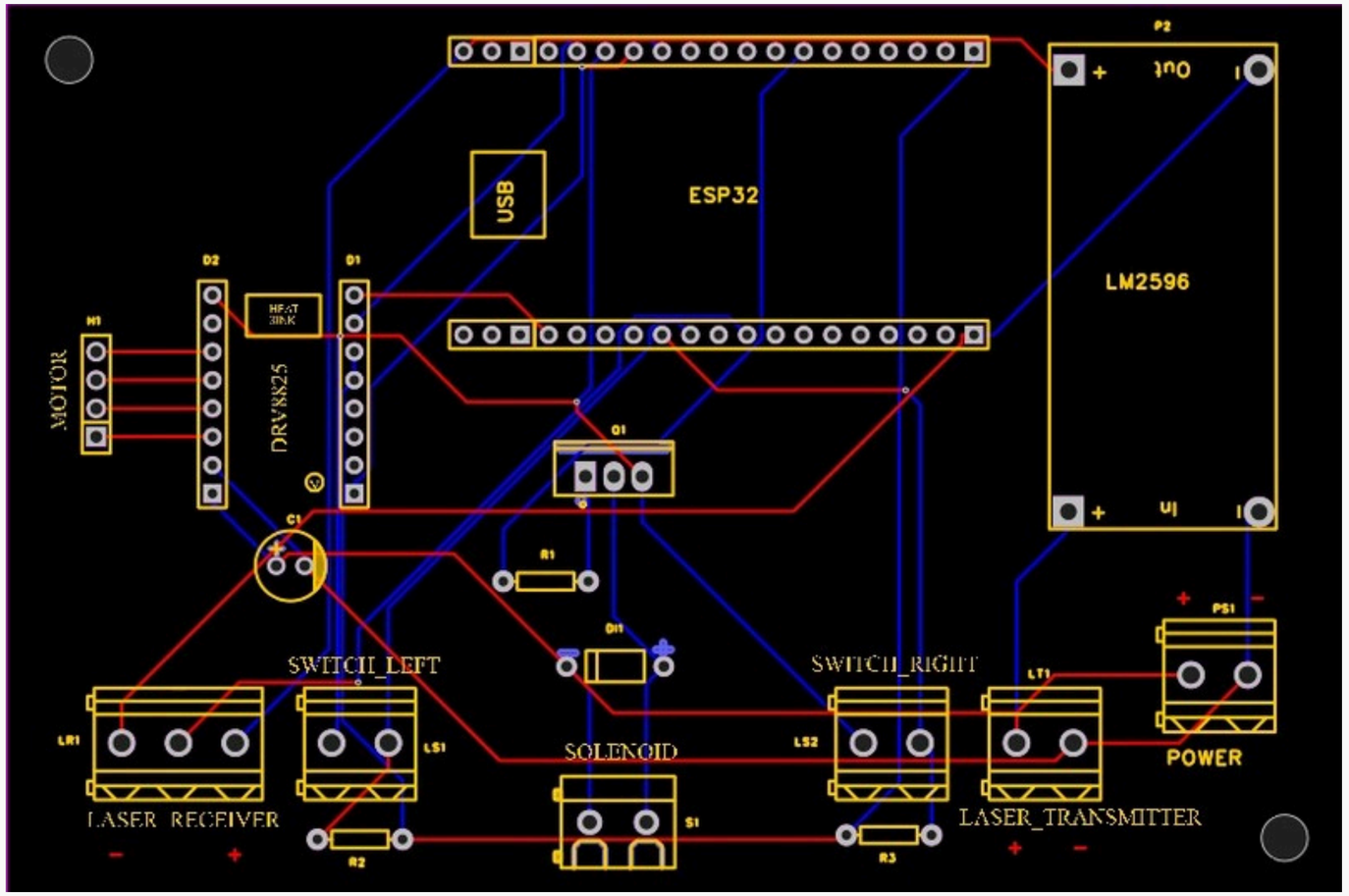


03 Robots need stable power, but batteries drain quickly during continuous play.

Constraints

PCB Design

04



We designed custom PCB layouts, but due to restrictions and geopolitical constraints, they were not fabricated.

BotStrike from the future

Mechanical Improvements-
Stronger motors and better wheels.

Use reinforcement learning so AI can improve over time.

Extended Gameplay Modes-
Multi-round tournaments and scoring systems.

THANK YOU !

Now it’s your turn to play