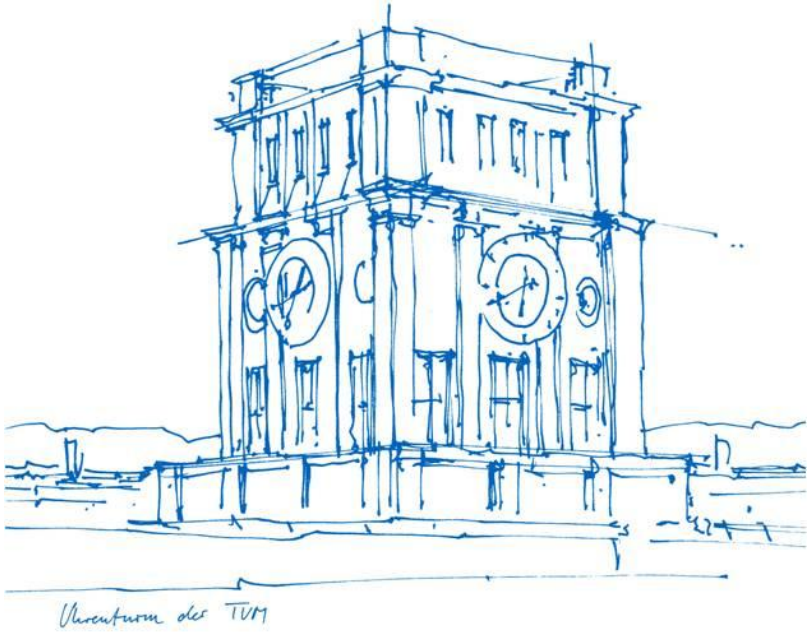
Air Hockey Robot

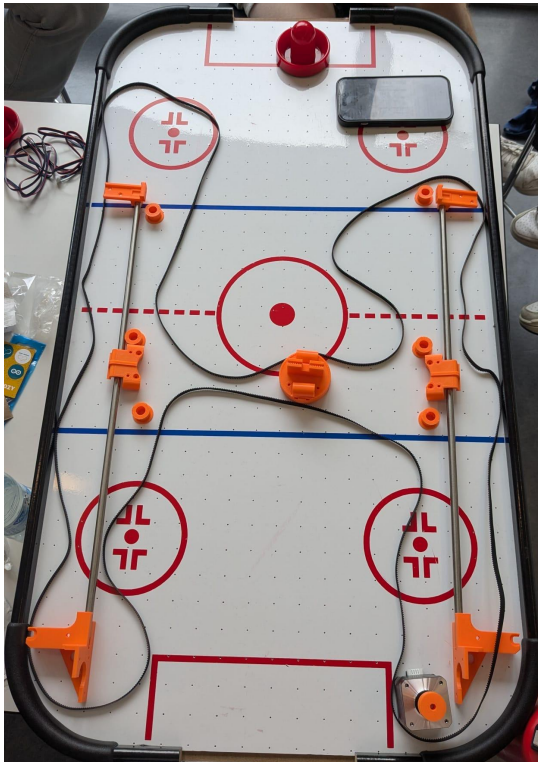
Felix Aker
Dzan Burzic
Gift Enabulele Asemota
Vrushabh Jain
Begüm Kara
Samuel Kemmler
Esmir Kićo



Introduction and Concept

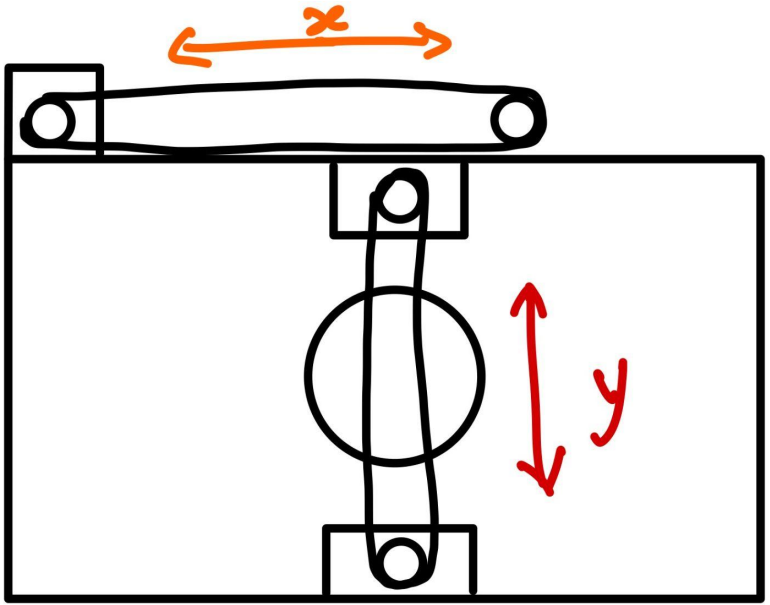
Construction of the robot

- Built on a rigid frame placed over the air hockey table.
- Linear guide rods added to support smooth carriage travel.
- Custom 3D-printed parts used for motor mounts, pulleys, and belt supports.
- Central carriage acting as the robot mallet.
- Continuous H-shaped belt routed through pulleys for precise motion.
- Assembly fastened with M3 screws for strength and alignment.



Robot Mechanics (Cartesian)

- In a **Cartesian** system, each axis (X and Y) is driven by its own independent motor, and the axes are stacked on top of each other.
- One axis carries the **weight** of the other, making movement heavier.
- **Slower acceleration** and direction changes.
- More stress on rails, leading to misalignment.



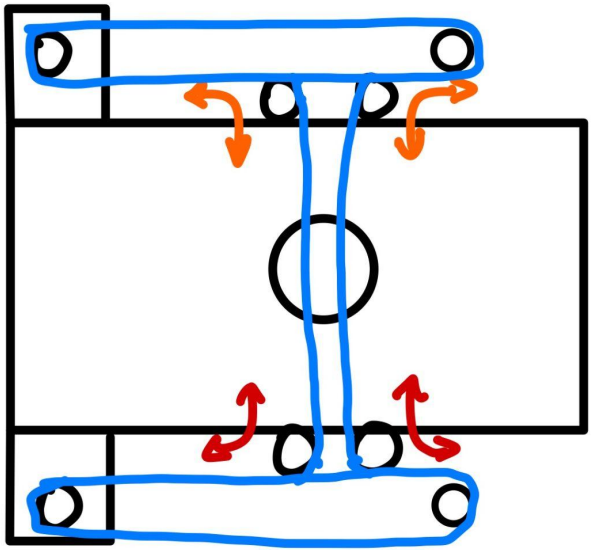
Robot Mechanics (H-Bot)

- The H-Bot uses two fixed motors connected by a single H-shaped belt to move a central carriage.

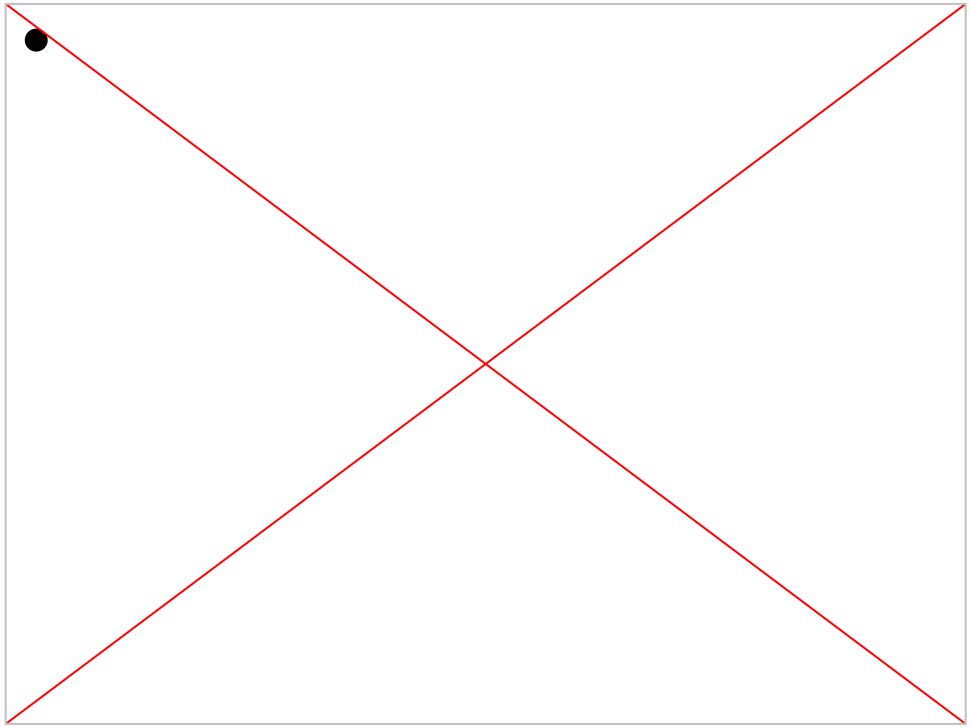
- Since both motors are fixed, no axis has to carry the weight of another, making the system lighter and more compact.

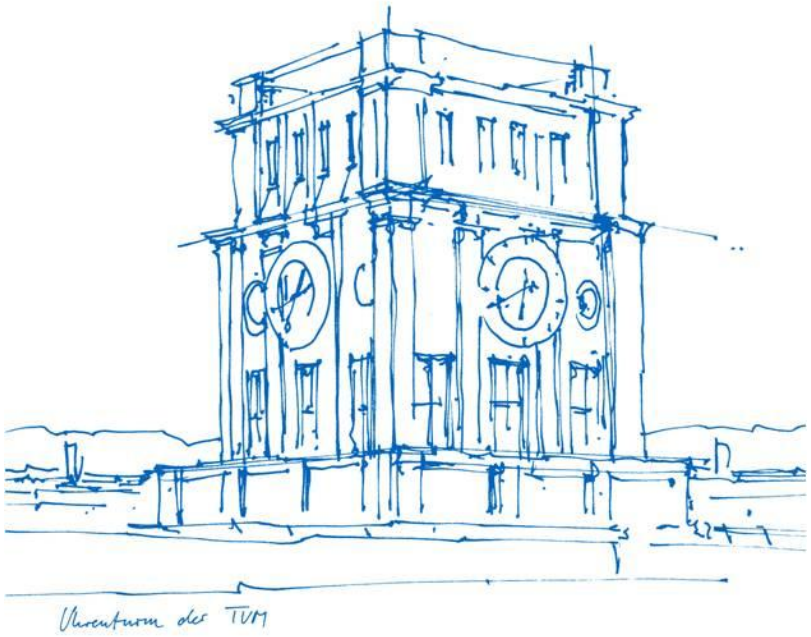
- The design is also simpler, using only two motors and a single continuous belt

- Both motors turning the same way → horizontal movement.
- Motors turning opposite ways → vertical movement.
- One motor moving → diagonal movement.



H-Bot Mechanics



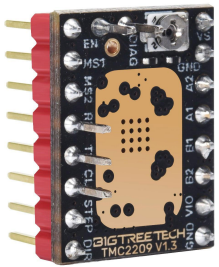


Electronic System

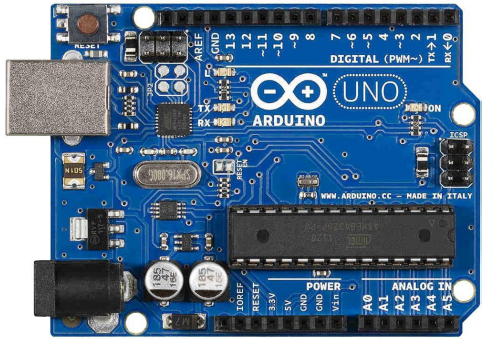
Goal:

- Develop reliable motor control system
- Ensure optimal performance and component longevity

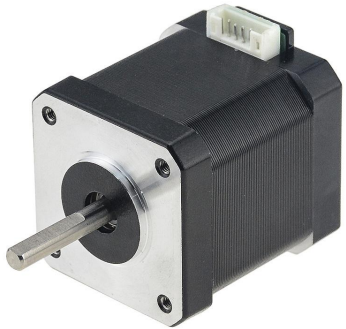
Electronic System: Components



TMC2209
Stepper Driver



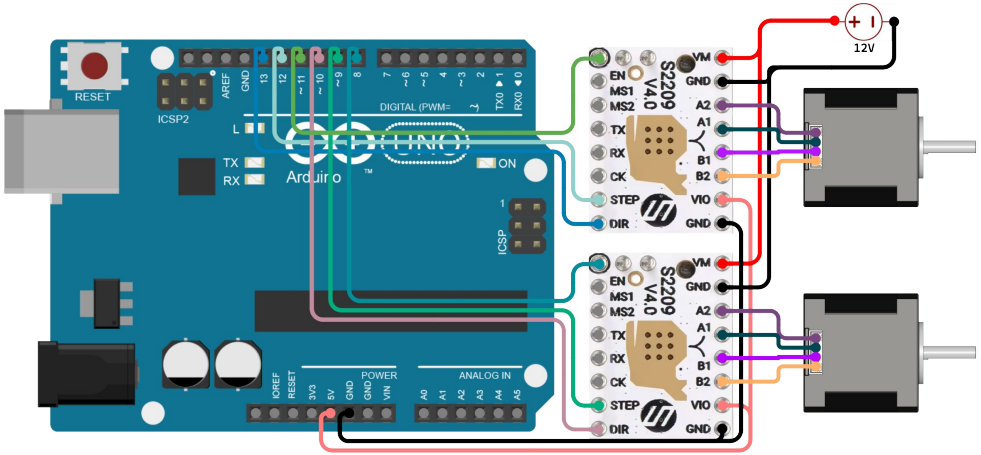
Arduino Uno
Microcontroller



Nema 17
Motor

Electronic System: Configuration

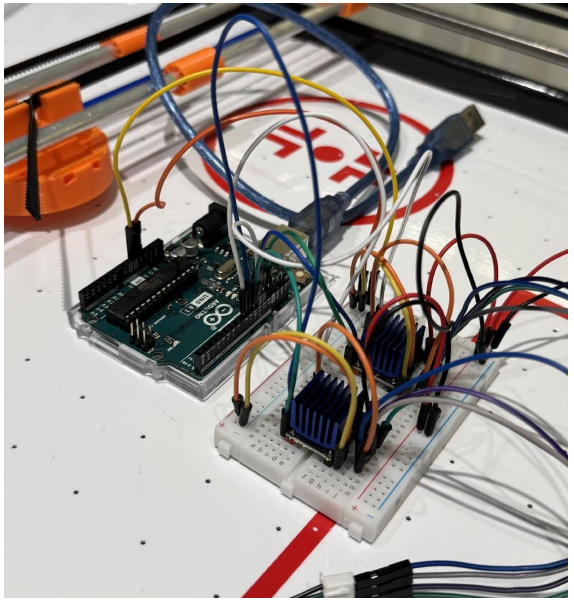
- The microcontroller sends signals for direction selection, step movement or pause via GPIO pins
- The coils of each motor connected to the output of the drivers
- The logical power provided by the microcontroller
- Motors powered by an external 12V power supply



Circuit Scheme

Electronic System: Current Limiting

- Current limited to achieve target motor rating of 1.7A
- Reference voltage set using analog calibration
- 80% power setting to prevent overheating and stressing components
- Long-term reliability improved while retaining most of the performance



Camera Vision System

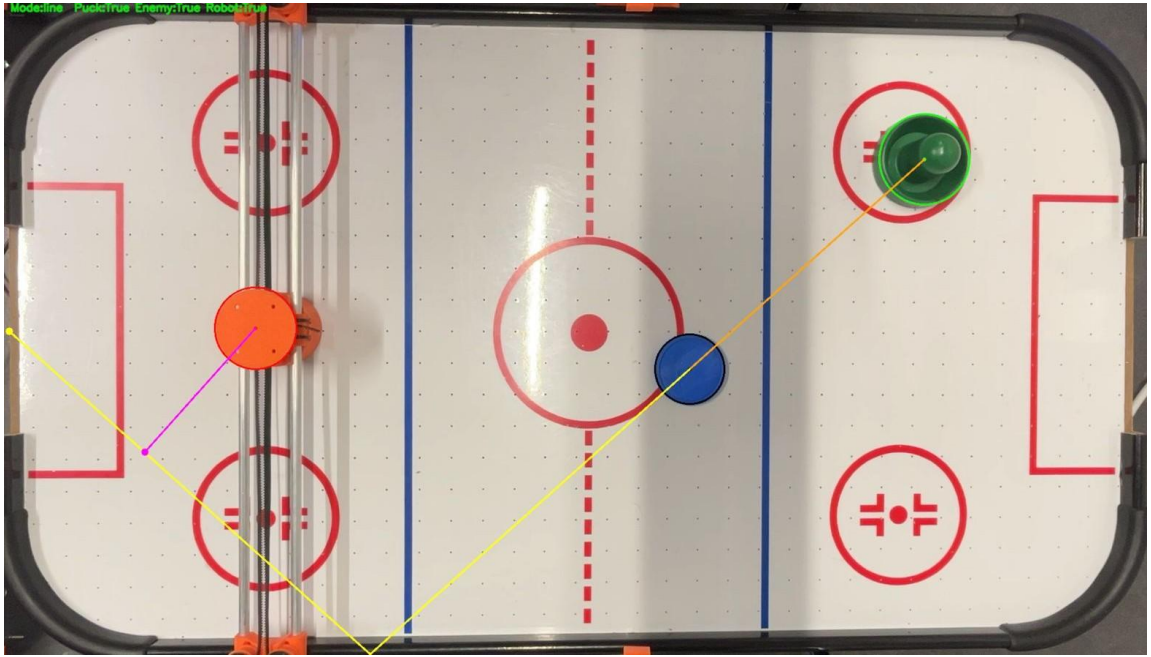
Goal:

- Track puck, human paddle, and robot paddle in real time.
- Output x- and y-position of all entities at any time

Camera Vision System

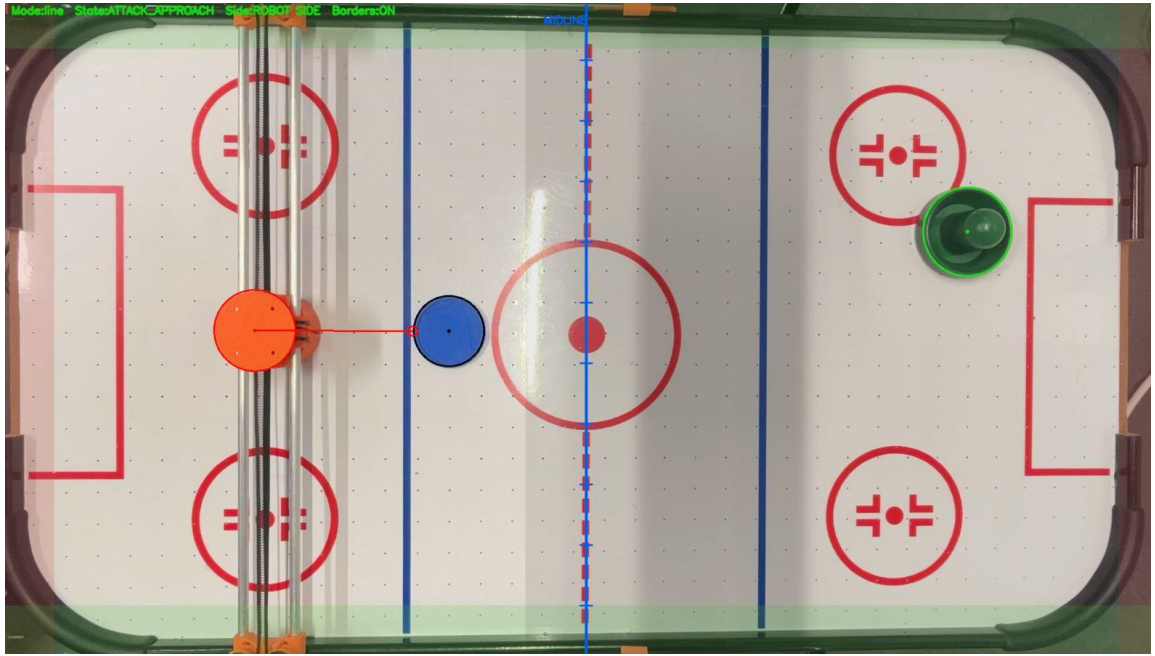
- Phone camera is installed on top of the table
- Each entity is marked with a unique color
- Converted image to HSV color space (better for color separation)
- Search in a different color range for each entity to differentiate between them
- Approximate circles using openCV
- Keeps last position if a object not visible for a moment

Camera Vision System



Green: Player paddle
Blue/Black: Puck
Orange/Red: Robot paddle
Yellow line: shot prediction
Pink line: needed movement to defend

Camera Vision System



Green: Player paddle
Blue/Black: Puck
Orange/Red: Robot paddle
Red line: needed movement to attack

Programming

Goal:

- Make defensive & offensive decisions
- Control robot precisely via Arduino

Programming

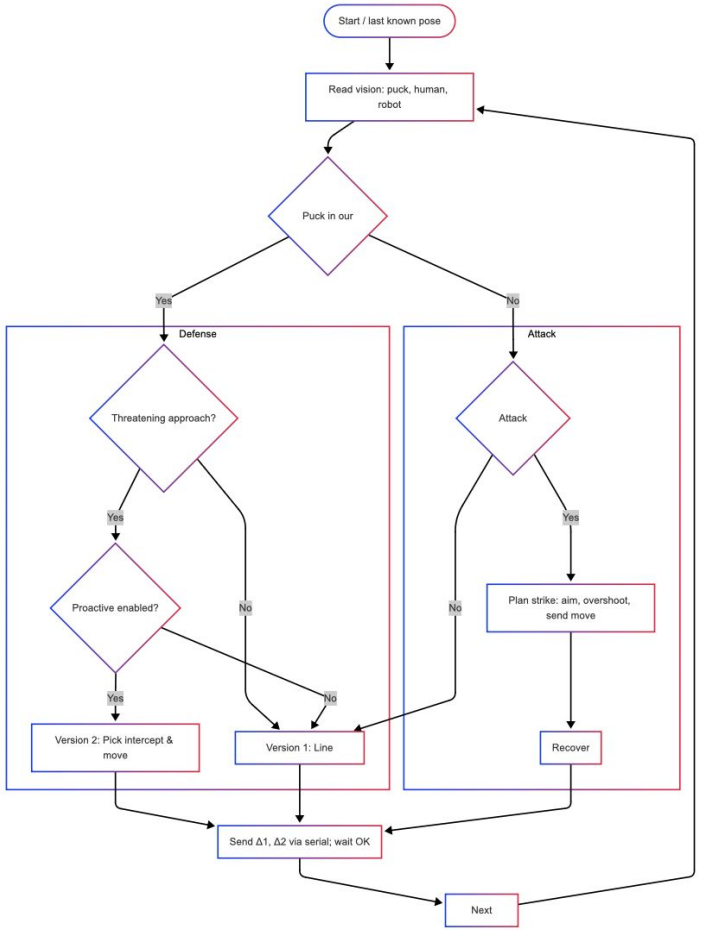
System Overview:

- Camera → Python Controller → Serial → Arduino → Motors

Three layers:

- Vision (input)
- Decision making (logic)
- Actuation (robot motion)

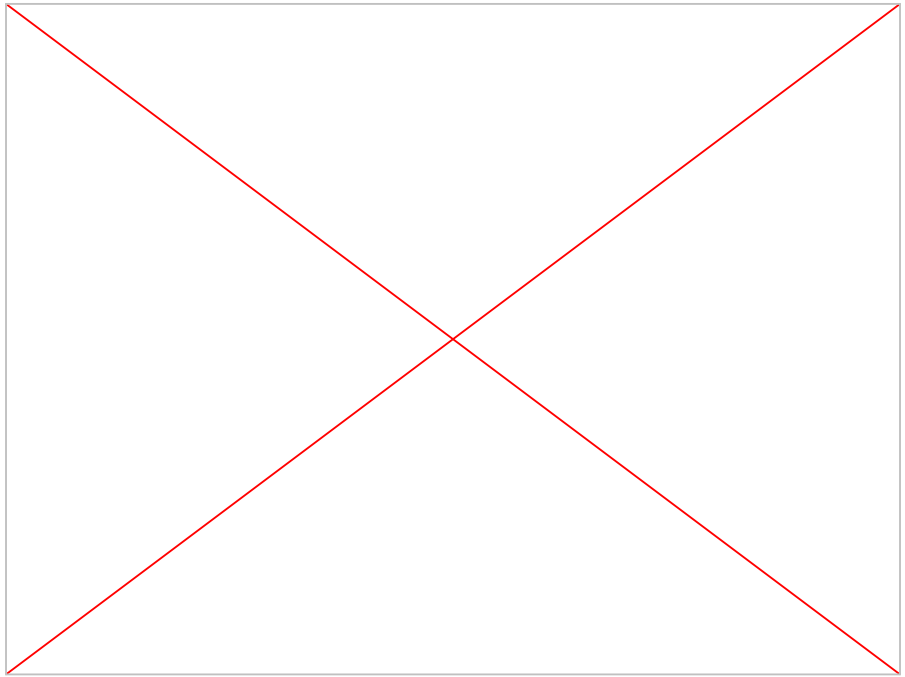
Algorithm


Start / last known pose
Read vision: puck, human, robot
Puck in our
Yes
No
Defense
Attack
Threatening approach?
Attack
Yes
Yes
Proactive enabled?
No
No
Plan strike: aim, overshoot, send move
Yes
No
Version 2: Pick intercept & move
Version 1: Line
Recover
Send Δ1, Δ2 via serial; wait OK
Next

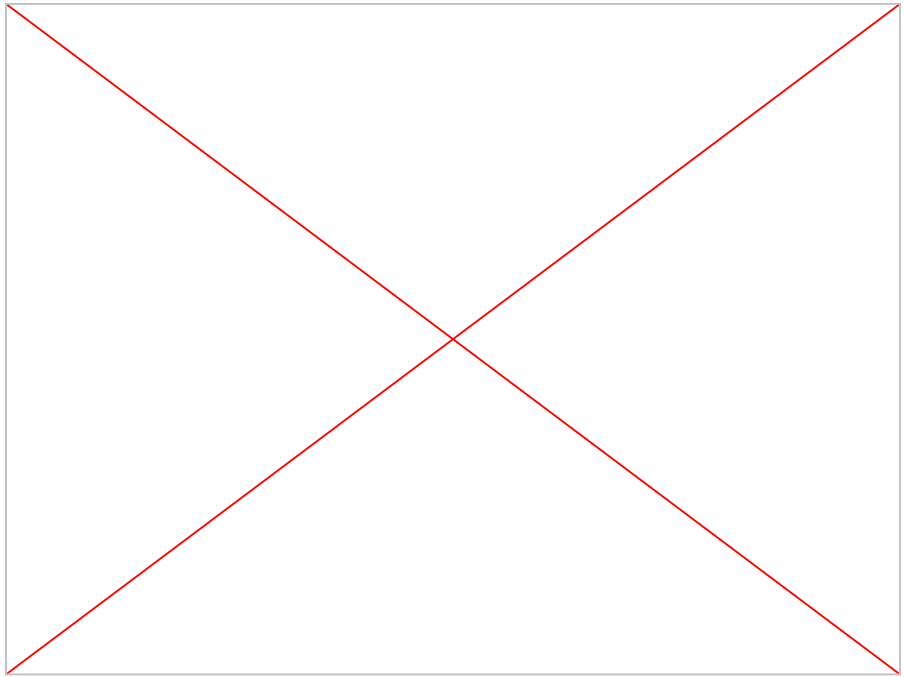
Coordinate System & CoreXY Mapping

- $\Delta 1 = \Delta x - \Delta y$
- $\Delta 2 = \Delta x + \Delta y$
- Benefits: simple firmware, Python does mapping

RobotVersion2.py (Predict puck path)



Robot.py (Line-based Defense)

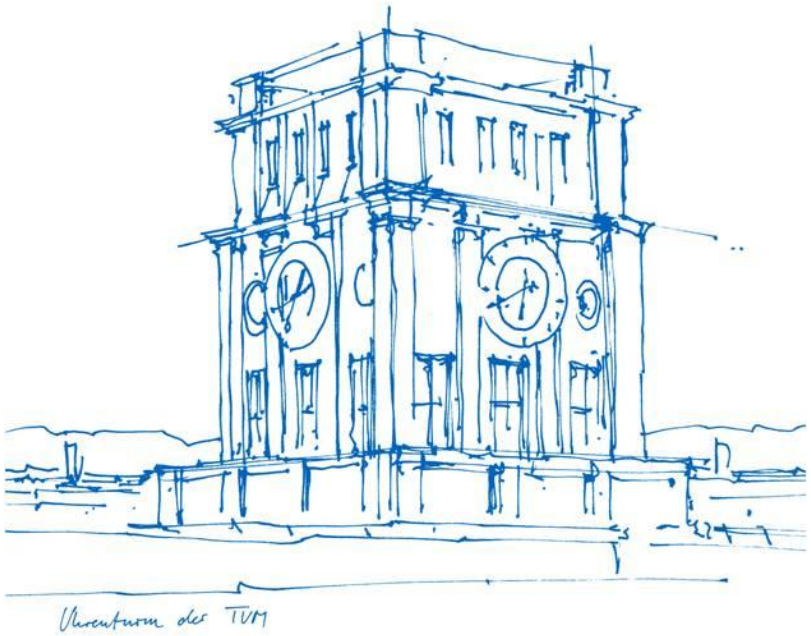


Arduino Firmware

- Pinout: STEP1/DIR1, STEP2/DIR2
- Motion block: interleaved stepping
- Acceleration ramp (limits stalls)
- Serial protocol: <delta1>,<delta2>\n ↔ OK
- Safety: 30s idle disables drivers

Future Improvement Opportunities

- Stronger and better built
- Stronger motor for better attack
- Add cooling fans for electronics
- Automatic goal counter



Thank you for listening…