

ECE 445
SENIOR DESIGN LABORATORY
FINAL REPORT

JengaBot: A Robotic System for Playing Jenga with a Human

Team #11

PEIRAN WEI
(pw19@illinois.edu)
WANGYIHAN GUO
(wguo12@illinois.edu)
JIACHENG YE
(ye32@illinois.edu)
HENG TIE ZHU
(hengtie2@illinois.edu)

Supervisor: Pavel Loskot
TA: Yue Yu

May 2026

Abstract

JengaBot is an autonomous robotic prototype for controlled Jenga block extraction in a human-robot game setting. It combines a two-dimensional motion platform, a single push-rod, a motorized rotating tower base, a multi-camera vision subsystem, and an embedded control architecture. The final demonstration uses a simplified 10-layer tower and does not require removed blocks to be re-stacked, allowing the project to focus on tower perception, turntable alignment, pusher positioning, and stable block removal.

The mechanical design evolved from a three-axis platform concept to a turntable-based architecture that presents different tower faces to a fixed pushing direction while the motion platform positions the pusher rod. AprilTag-based visual identification supports block localization and tower-state reconstruction, with a computer handling high-level perception and strategy and an Arduino-based controller managing low-level motor commands. Mechanical integration included CAD modeling, 3D-printed parts, platform installation, acrylic-base mounting, and alignment. The final prototype demonstrates the feasibility of a compact robotic block-pushing system and provides a foundation for future improvements such as force feedback, robust perception, and complete pick-and-place gameplay.

Contents

1	Introduction	1
1.1	Problem Statement	1
1.2	Solution Overview	1
1.3	High-Level Requirements	2
1.4	Block-Level Changes from Proposal	3
1.5	Weekly Schedule	4
2	Design	6
2.1	Design Procedure	6
2.2	Design Details	7
2.2.1	Overall Structure	7
2.2.2	Vision Subsystem	7
2.2.3	Motion and Turntable Subsystem	9
2.2.4	Operation Unit: Push-Rod End-Effector	11
2.2.5	Central Controller	12
2.2.6	Power Supply	12
2.3	Tolerance Analysis	13
3	Verification	14
3.1	Vision Subsystem Verification	14
3.2	Motion and Turntable Subsystem Verification	14
3.3	Push-Rod Operation Unit Verification	15
3.4	Central Controller Verification	15
3.5	Power Supply Verification	16
3.6	System-Level Verification	16
4	Costs	17
4.1	Labor Costs	17
4.2	Parts Costs	17
4.3	Mass-Production Cost Estimate	18
5	Conclusions	19
5.1	Accomplishments	19
5.2	Uncertainties and Future Work	19
5.3	Ethical Considerations	20
5.4	Safety	20
5.5	Broader Impacts	20
	References	22
	Appendix A Code Repository Structure	23
	Appendix B Requirement and Verification Table	24

1 Introduction

This report presents JengaBot, an autonomous robotic prototype for simplified Jenga block extraction against a human opponent. The project demonstrates precise object manipulation in a confined workspace using 3D-printed structural components, a multi-camera vision system, a two-dimensional motion platform, a motorized rotating tower base, and a single push-rod end-effector.

1.1 Problem Statement

While 3D printing and embodied intelligence have progressed significantly, precise object manipulation in constrained spaces remains complex. Conventional robotic arms are often too bulky for small tabletop workspaces, whereas customized 3D-printed structures can provide a compact, task-specific mechanical layout. This project explores combining a compact 3D-printed frame with precision motion hardware to perform delicate block-pushing manipulation within the limited space around a Jenga tower.

The mechanical challenge of JengaBot differs from a purely software-based game system. The robot must physically interact with the tower, approach the correct block, align a pushing tool with the target face, and apply controlled lateral motion without disturbing the tower excessively. Therefore, the project focuses not only on perception and decision-making, but also on mechanical reachability, turntable stability, pusher alignment, and reliable prototype assembly.

1.2 Solution Overview

To address workspace limitations, JengaBot integrates a custom 3D-printed frame, a two-dimensional linear motion platform, a motorized rotating tower base, and a single push-rod operation unit. Placing the tower on a motorized turntable allows different faces to be presented to a fixed pushing direction, reducing end-effector complexity and making the mechanical design achievable for a semester prototype.

Four OV9732 cameras **ov9732** mounted at the base capture the tower state from multiple angles. The central control software processes these images, constructs or updates a digital tower representation in MuJoCo simulation [1], selects a target block, and sends motion commands to the mechanical subsystem. An Arduino Uno R3 **arduino-uno** serves as the low-level motion controller, receiving commands from the central controller and generating STEP signals for the motion platform, TMC2209 stepper motor driver **tmc2209**, and DC pushrod.

The system comprises five major subsystems: (1) vision, including cameras and AprilTag detection; (2) motion and turntable, including the two-dimensional platform and rotating base; (3) operation unit, implemented as a push-rod end-effector; (4) central controller, including image-processing and strategy software; and (5) power supply.

Fig. 1 shows the system block diagram with data and power connections among these subsystems. Fig. 2 shows the CAD model of the integrated mechanical assembly.

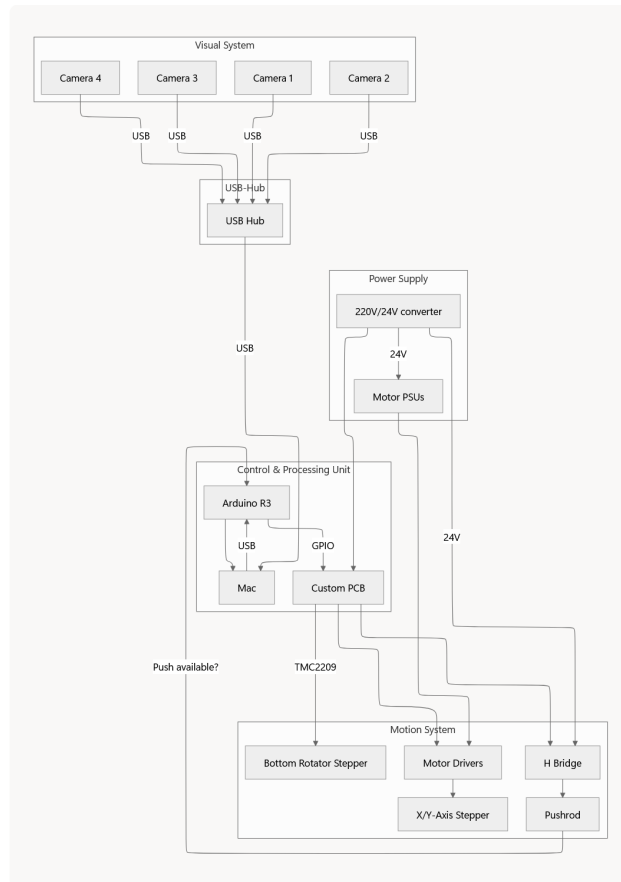


Figure 1: System design block diagram

1.3 High-Level Requirements

The system is designed to meet the following high-level performance requirements for the simplified JengaBot demonstration:

1. **Core Manipulation Function:** The system shall autonomously complete the main JengaBot interaction sequence: detecting or updating the tower state, selecting a target block, rotating the tower to the required orientation, positioning the push rod with respect to the selected block, and performing a controlled pushing action without causing immediate tower collapse due to robot-induced error. For the final demonstration, the system uses a simplified 8-layer tower and does not require removed blocks to be placed back on the top layer.
2. **Mechanical Positioning and Reachability:** The motion platform, rotating base, and push-rod assembly shall reach the useful block-height range of the 10-layer tower. The pusher shall approach the target block along a straight and repeatable path without colliding with the frame, turntable, camera region, or surrounding mechanical components. The target positioning accuracy is ± 1.0 mm within the effective operating region, which is sufficient for aligning the pusher with individual Jenga blocks of size $30 \text{ mm} \times 30 \text{ mm} \times 90 \text{ mm}$.

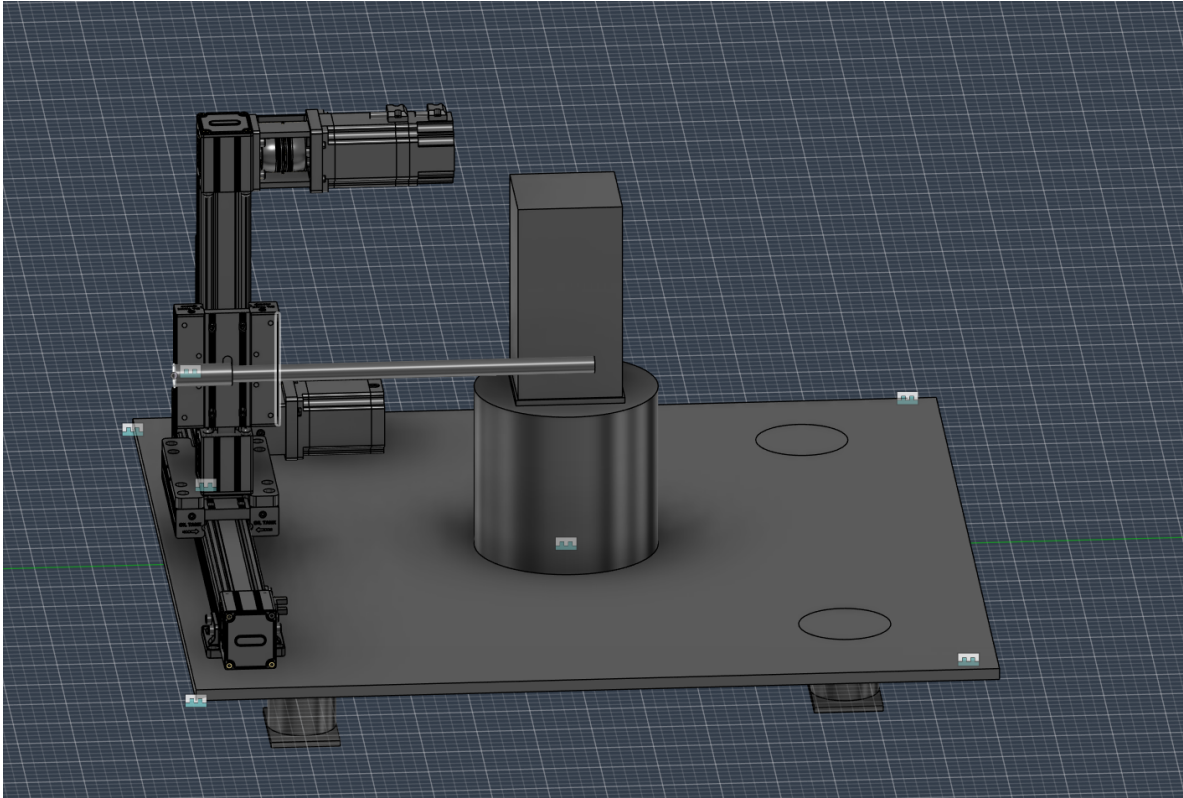


Figure 2: Integrated CAD layout of the JengaBot prototype, showing the two-dimensional motion platform, single push-rod end-effector, square frame region, and motorized rotating tower base.

3. **Integrated Demonstration Time:** The system shall complete one simplified manipulation cycle, including image capture, tower-state update, target selection, tower rotation, pusher positioning, and controlled block pushing, within 15 s under normal prototype demonstration conditions. This requirement focuses on the extraction and pushing sequence rather than a full official Jenga cycle with block re-stacking.

1.4 Block-Level Changes from Proposal

Since the original proposal, the mechanical subsystem underwent a significant revision. The initial direct two-axis motion platform concept proved impractical for pushing blocks from all sides, so the team adopted a turntable-based architecture: the tower sits on a motorized rotating base while a two-dimensional motion platform positions a single push rod relative to the selected block.

This change reduces operation-unit complexity. Rather than building multiple pushing mechanisms or a larger manipulator, the robot rotates the tower to expose different faces to one fixed pushing direction. The purchased two-dimensional platform reduces fabrication risk compared with scratch-built axes, while allowing the team to focus on custom

integration of the rotating base, push-rod mounting, 3D-printed connectors, acrylic-base layout, and final assembly.

A second important change is the simplification of gameplay rules. A complete official Jenga-playing robot would require a complicated gripping and lifting mechanism to re-stack blocks. The final course prototype uses a 10-layer tower and does not re-stack removed blocks, and stable mechanical operation.

1.5 Weekly Schedule

Table 1: Weekly Task Allocation

Week	Wangyihan Guo	Peiran Wei	Jiacheng Ye	Hengtie Zhu
3.2	Find correct simulator	Checkout camera arguments	Develop initial mechanical concept for frame and motion layout	Develop initial mechanical concept for frame and motion layout
3.9	Find platform	Build up recognizing system demo	Build CAD concept for 3D-printed frame and motion mounting structure	Build CAD concept for 3D-printed frame and motion mounting structure
3.16	Buy and test Pi5	Validate system robustness on single camera	Selection of XY-axis motion platform	3D diagram modeling
3.23	3D reconstruction	Determine detailed Jenga brick size and label accuracy	3D diagram modeling	Integrate end-effector with Z-axis carriage and verify basic mechanical motion
3.30	Try to simulate tower	Validate system robustness on all cameras	Turntable design	Turntable design

Continued on next page

Table 1 – *Continued from previous page*

Week	Wangyihan Guo	Peiran Wei	Jiacheng Ye	Hengtie Zhu
4.6	Finish simulator building	Use 3D-printed substitution bricks to validate latency	Selection of push rod actuator	Selection of push rod actuator
4.13	Test motors	Develop demo control algorithm in simulation	Fixing component design	Fixing component design
4.20	Print PCB board	Develop demo control algorithm in simulation	3D printing	3D printing
4.27	Test 3-axis platform	Use 3D-printed substitution bricks to validate/tune algorithm	Align vertical axis and check mechanical mounting	Align vertical axis and check mechanical mounting
5.4	Apply running algorithm	Tune algorithm on real Jenga bricks	Measure travel range and preliminary repeatability	Measure travel range and preliminary repeatability
5.11	—	Tune algorithm on real Jenga bricks	First block-contact and extraction test	First block-contact and extraction test
5.18	Final Report	Final Report	Final Report	Final Report

2 Design

The system is organized into five major subsystems: vision, motion and turntable control, push-rod operation, central control, and power supply. This section describes the design rationale and detailed implementation of each subsystem, with special emphasis on the mechanical architecture used to make the simplified Jenga demonstration achievable.

2.1 Design Procedure

The system architecture follows a perception–decision–action pipeline common to autonomous robotic systems. The final design was selected after comparing several alternative approaches in mechanical structure, end-effector operation, control architecture, and vision method.

Centralized vs. distributed control: A fully centralized architecture with the Mac handling both high-level computation and real-time motor pulse generation was considered. This approach was rejected because Linux-based systems can introduce timing jitter in motor pulse trains, which may degrade stepper motor performance. Instead, a split-controller architecture was adopted. The Mac handles image processing, tower-state reconstruction, and strategy computation, while the Arduino Uno R3 generates deterministic STEP/DIR signals for the stepper motor drivers.

Three-axis platform vs. turntable-based architecture: The initial mechanical concept considered using a direct three-axis motion platform to approach the Jenga tower from outside the frame. During design review, this approach was found to be less practical because a fixed platform would have difficulty pushing blocks from all sides of the tower. The final design therefore uses a turntable-based architecture. The Jenga tower is placed on a motorized rotating base, and the push rod approaches from a fixed direction. By rotating the tower, different tower faces can be presented to the pusher without requiring a large robotic arm or multiple pushing mechanisms.

Push-only operation vs. full pick-and-place manipulation: A full official Jenga-playing robot would need to remove a block and place it back on the top layer. This would require a more complicated gripping, lifting, and re-stacking mechanism. For the final course prototype, the team simplified the game rule by using a 10-layer tower and not placing removed blocks back on the top. This allows the mechanical design to focus on the most important prototype goal: rotating the tower, aligning the pusher, and performing controlled block extraction.

Custom fabrication vs. purchased motion platform: Building all linear motion axes from scratch would create risks in rail alignment, stiffness, backlash, and repeatability. The team therefore used a purchased two-axis motion platform as the main positioning structure and integrated it with custom 3D-printed parts, a motorized turntable, and a push-rod operation unit. This decision reduced fabrication risk while still leaving significant mechanical design work in platform mounting, pusher alignment, turntable integration, and final assembly.

Vision approach: A depth-camera-based approach, such as using an Intel RealSense camera **realsense**, was considered for tower-state reconstruction. However, the team selected an AprilTag-based fiducial marker approach [2] because AprilTags provide direct block ID information and are easier to debug than complex point-cloud segmentation. The tag-based method supports multi-view block localization and can be integrated with the digital tower model used by the strategy algorithm.

2.2 Design Details

2.2.1 Overall Structure

The overall structure of the JengaBot system is shown in Fig. 3. The system is built around a perception–decision–action workflow. In the perception stage, multiple cameras mounted around the Jenga tower capture images from different viewpoints and detect AprilTags attached to the blocks. The detected tag information is then transmitted to the Mac, which reconstructs or updates the tower state, maintains the digital representation of the game scene, and determines the next block to be pushed.

After the target block is selected, the Mac sends motion commands to the Arduino Uno R3 through a USB serial connection. The Arduino generates STEP/DIR control signals for the TMC2209 stepper motor driver boards **tmc2209**. These drivers control the motion platform, the rotating tower base, and the push-rod operation unit. Mechanically, the system relies on a turntable-based layout: the tower rotates to expose the desired face, while the pusher approaches from one controlled direction.

The system also includes an independent power supply module. A 24 V switching power supply provides power for the stepper motor drivers and motors, while the control electronics are powered separately to improve system stability. Through the integration of visual perception, motion control, turntable alignment, push-rod operation, central control, and power management, the system forms a compact robotic platform for simplified Jenga block extraction.

2.2.2 Vision Subsystem

Four OV9732 cameras are mounted around the frame to capture the Jenga tower from multiple viewpoints. Each camera provides an independent image stream for AprilTag detection. The multi-camera layout improves the chance that at least one visible tag can be detected on each candidate block, even when some faces are blocked by the tower geometry or by the pusher structure.

Each Jenga block carries AprilTags from the tag36h11 family [2]. The tags provide direct block ID information, which helps the software determine which blocks remain in the tower and where the candidate blocks are located. Table 2 summarizes the AprilTag placement convention used for each block.

Each camera runs an independent AprilTag detection pipeline using the `apriltag` Python library [2]. The detector identifies each tag’s ID, pixel center, and four corner coordinates.

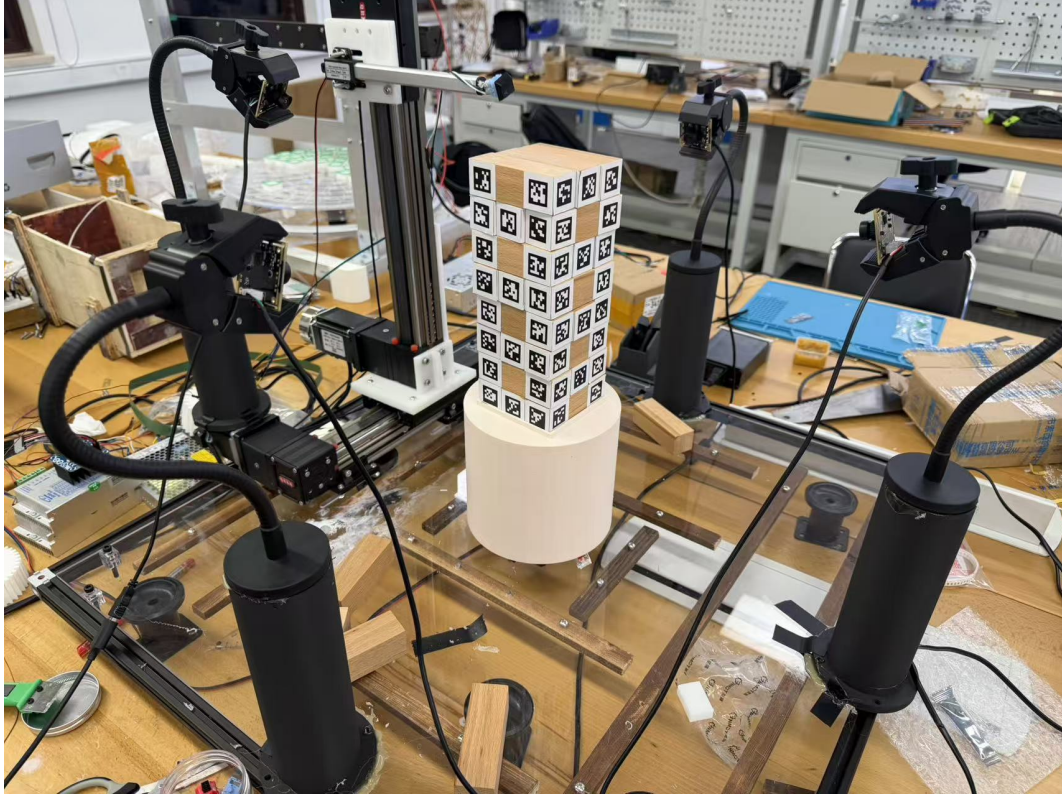


Figure 3: Overall structure of the JengaBot system.

Table 2: AprilTag placement convention for each Jenga block.

Index	Suffix	Face	Relative Position
0	end_pz	+Z end	$(0, 0, +\text{END_Z})$
1	end_nz	-Z end	$(0, 0, -\text{END_Z})$
2-3	x_head, x_tail	+X side	$(\pm\text{SIDE_XY}, 0, \pm 1.0)$
4-5	nx_head, nx_tail	-X side	$(\mp\text{SIDE_XY}, 0, \pm 1.0)$
6-7	y_head, y_tail	+Y side	$(0, \pm\text{SIDE_XY}, \pm 1.0)$
8-9	ny_head, ny_tail	-Y side	$(0, \mp\text{SIDE_XY}, \pm 1.0)$

The detected information is then sent to the central controller for tower-state reconstruction and target selection.

For calibration, fixed AprilTags are mounted on the turntable or base region as reference markers. At system startup, the cameras use these reference tags to estimate their relative positions and improve consistency between the physical tower and the digital scene. This calibration requirement also affects the mechanical design because the frame, pusher, and turntable should not block the cameras' view of the reference tags.

2.2.3 Motion and Turntable Subsystem

The motion subsystem consists of a purchased two-axis motion platform, a motorized rotating tower base, and a push-rod mounting structure. The platform positions the pusher relative to the tower, while the turntable rotates the Jenga tower so that different tower faces can be accessed from a single pushing direction. This design avoids the need for a bulky robotic arm and makes the prototype more compact. Although the CAD model includes a vertical carriage for setting the pusher height, the final demonstration does not rely on a full pick-and-place three-axis gantry. The vertical adjustment is mainly used to align the push rod with the selected block layer in the simplified 10-layer tower.

Fig. 4 shows the detailed CAD model of the motion platform and vertical carriage used during the mechanical design stage. This model was used to check pusher mounting, travel range, carriage clearance, and the relationship between the platform and the tower workspace. It also helped the team identify practical assembly concerns before final mounting, such as screw access, motor clearance, carriage interference, and the need to keep the push rod aligned with the selected block height.

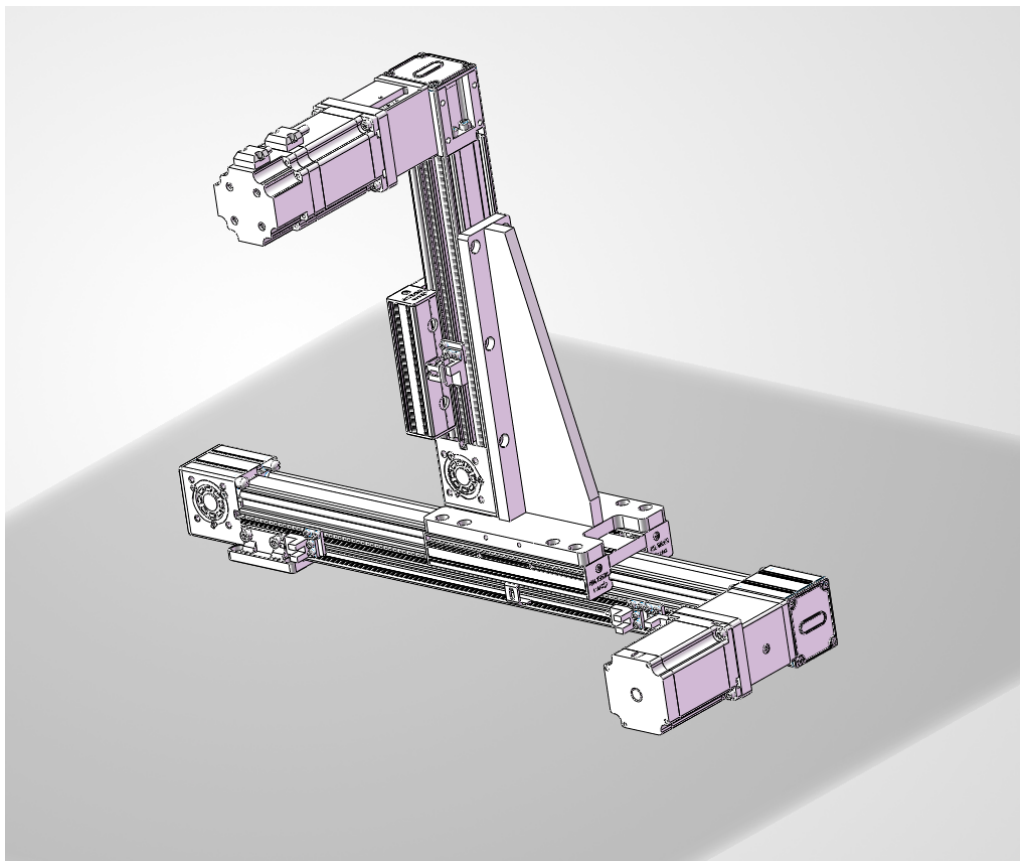


Figure 4: Detailed CAD model of the motion platform and vertical carriage used to plan the pusher mounting position, travel path, and mechanical clearance.

The turntable is the key mechanical change from the original platform-only concept. In a direct platform design, the pusher would have difficulty accessing all four sides of the

tower. In the final design, the tower is rotated instead. This means the pusher can remain mechanically simple, while the tower orientation is adjusted to present the selected block face to the push rod.

Fig. 5 shows a CAD close-up of the motorized turntable used in the final mechanical design. This model was used to inspect the circular support geometry, central shaft interface, and mounting-hole arrangement before physical fabrication and assembly.

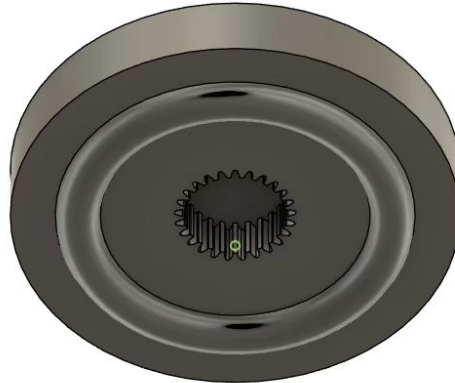


Figure 5: CAD close-up of the motorized turntable used to support and rotate the simplified Jenga tower. The model shows the circular support plate, central shaft region, and mounting-hole layout used for turntable integration.

The turntable must satisfy two main mechanical requirements. First, it must keep the Jenga tower close to the center of rotation so that the block position remains predictable after rotation. Second, it must rotate smoothly without creating excessive wobble or shifting the tower before the pushing motion begins. Since the turntable directly affects the relative position between the tower and the push rod, centering and rotational stability are more important than high rotational speed for this prototype.

The turntable parts were designed in CAD and fabricated with 3D printing. Because printed parts may have dimensional errors, the design allowed for post-processing such as drilling, hole cleanup, and fit adjustment. This was important for the shaft interface, screw holes, and mounting locations, where small errors could create eccentric rotation or assembly misalignment.

The final mechanical layout uses an acrylic base plate to support the motion platform and turntable. Since the purchased motion platform is relatively heavy, the base must provide enough stiffness to avoid visible deflection during movement. The assembly process included drilling holes, checking screw alignment, using M5 fasteners for major mounting points, and adjusting printed connectors when tolerances did not match the CAD model exactly. These physical fit checks were necessary to make sure that the pusher

path, turntable location, camera region, and wiring space could all coexist in the same compact workspace.

2.2.4 Operation Unit: Push-Rod End-Effector

The operation unit uses a push-rod end-effector rather than a full hybrid gripper, and the design of the end-effector is shown in fig. 6. The push rod is mounted on the moving platform and aligned with the centerline of the selected Jenga block before each extraction attempt. During operation, the system first rotates the tower to the required orientation, then positions the pusher, and finally applies a controlled lateral displacement to the target block.

This push-focused design is consistent with the simplified demonstration rule. Since removed blocks are not placed back on the top layer during the final prototype demonstration, the system does not need a lifting and re-stacking gripper. Removing the re-stacking step reduces mechanical complexity, lowers the moving mass of the end-effector, and makes the system easier to integrate within the available workspace.

The main design constraints for the push rod are reachability, straight-line approach, clearance, and disturbance control. The pusher must reach the useful block-height range of the 10-layer tower, approach the target block without colliding with the frame or camera region, and move slowly enough to avoid sudden impact loading on the tower. The current prototype does not use closed-loop force sensing, so the pushing strategy relies on conservative displacement, low-speed motion, and limited stroke length.



Figure 6: Pushrod unit of the JengaBot system.

2.2.5 Central Controller

The central controller is split between the Mac and the Arduino Uno R3 **arduino-uno**. The Mac receives multi-camera image data, runs the perception and strategy pipeline, and sends target commands to the Arduino through USB serial communication. The Arduino handles low-level motor control by generating deterministic STEP/DIR signals and reporting execution status back to the Mac.

This split-controller architecture isolates high-level processing from low-level pulse generation. As a result, image processing or strategy computation on the Mac does not directly interfere with the timing of motor control. This is especially important for stepper motor operation, where unstable pulse timing can lead to vibration, missed steps, or inconsistent motion.

The picture of the controller is shown in Fig. 7.

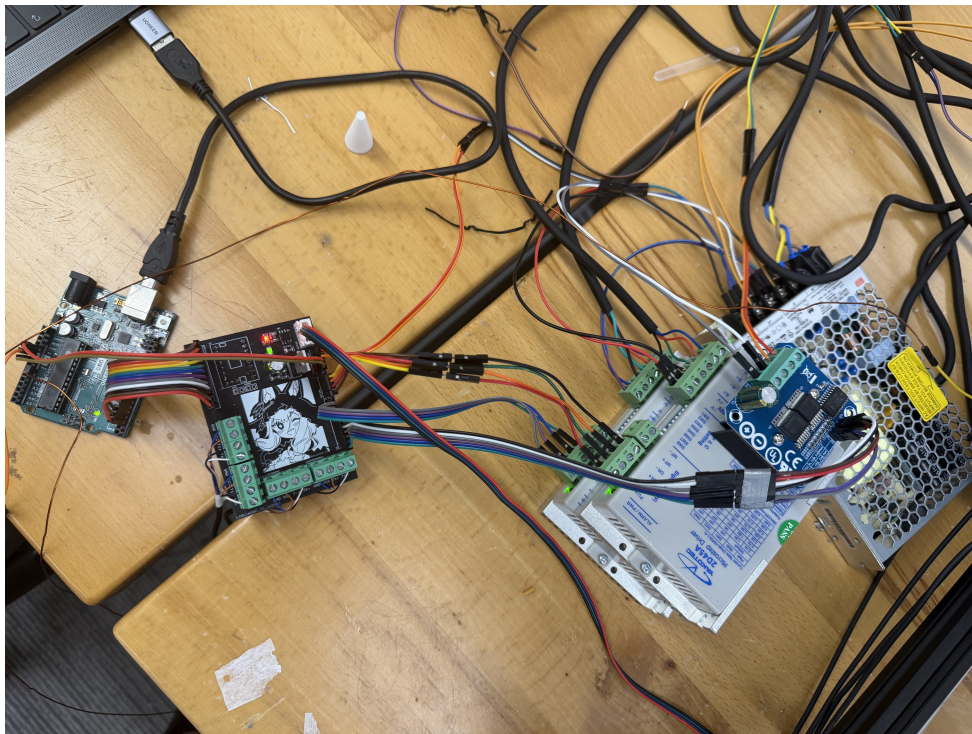


Figure 7: Control unit of the JengaBot system.

2.2.6 Power Supply

The motor drive stage is powered by a 24 V / 6 A switching power supply, which provides power to the TMC2209 driver boards and stepper motors. The control electronics are powered separately to reduce interference between motor current transients and logic-level circuits. Each motor driver board includes local capacitance near the motor supply input to suppress voltage fluctuation during acceleration, deceleration, and start-stop motion.

2.3 Tolerance Analysis

Two critical mechanical questions were evaluated analytically: whether the motion platform has enough positioning resolution to align with an individual Jenga block, and whether the push-rod force is suitable for prototype-level block extraction.

Positioning Accuracy: Each Jenga block has nominal dimensions of 30 mm × 30 mm × 90 mm. For a standard NEMA17 stepper motor with a 1.8° step angle, 200 full steps per revolution, 16× microstepping, and an estimated 40 mm axis travel per revolution, the theoretical linear resolution is:

$$\text{Linear resolution} = \frac{40 \text{ mm/rev}}{200 \times 16 \text{ microsteps/rev}} = 0.0125 \text{ mm/microstep} \quad (1)$$

This theoretical resolution is much finer than the target system-level positioning tolerance of ±1.0 mm. In practice, the dominant errors are not caused by microstep resolution, but by mechanical factors such as frame flexibility, backlash, rail alignment, printed-part tolerance, screw-hole clearance, and base deflection. Relative to the 30 mm block width, a ±1.0 mm positioning error is approximately 3.3%, which leaves enough margin for aligning the push rod with the central region of the block face.

Turntable Alignment: The turntable must keep the Jenga tower close to the intended center of rotation. If the tower is placed off-center, rotating the base will shift the block face relative to the pusher path, reducing alignment accuracy. Therefore, the turntable design should provide a clear reference for tower placement, and the pusher path should be checked after rotation. For the prototype, visual alignment marks and repeated fit checks are sufficient. A future version could use a more precise centering fixture or a bearing-supported turntable to reduce eccentric motion.

Push Force Feasibility: The required extraction force depends on block-to-block friction. A simplified Coulomb friction model gives:

$$F = \mu N \quad (2)$$

where μ is the effective static friction coefficient between Jenga blocks and N is the normal force from the upper layers. The exact extraction force varies from block to block because of tower loading, surface condition, block manufacturing tolerance, and local contact pressure. Therefore, the push rod should not rely on a sudden high-force impact. Instead, the prototype uses a controlled and low-speed pushing motion to reduce tower disturbance.

The tolerance analysis shows that the motor resolution is sufficient for the target alignment task, but the real performance depends mainly on mechanical stiffness, turntable centering, and careful pusher setup. The push-rod concept is suitable for the simplified prototype, while future versions should add force feedback, current monitoring, or a compliant pusher tip to improve safety and repeatability.

3 Verification

This section discusses prototype-level verification of the completed JengaBot system and its major subsystems. The full Requirement and Verification Table is provided in Appendix B. Since the final demonstration follows a simplified Jenga scope, verification focuses on whether the system can observe the tower, rotate it, position the push rod, and perform a controlled block-pushing action. Removed blocks need not be transported or re-stacked.

3.1 Vision Subsystem Verification

The vision subsystem was verified by checking whether the camera arrangement could provide usable image input for AprilTag-based tower observation. Each camera was connected to the central controller, and sample frames were captured to confirm that the Jenga tower and visible AprilTag surfaces could be observed from the assigned viewpoints.

The AprilTag detection pipeline was tested on sample tower images. The software returned tag IDs, image centers, and corner coordinates for visible tags, providing the required information for block identification and tower-state reconstruction. The multi-camera layout improved visual coverage compared with a single fixed viewpoint because different sides of the tower could be observed from different camera positions.

Fixed reference tags on the base or turntable region were used to support workspace alignment and calibration. These tests showed that the cameras and tag-detection software can provide the necessary tower-state information for the simplified prototype demonstration. The main remaining limitation is that tag detection can still be affected by lighting, camera placement, and occlusion from the tower or pusher structure.

3.2 Motion and Turntable Subsystem Verification

The motion and turntable subsystem was verified according to the final turntable-based architecture. Verification focused on the purchased two-axis motion platform, the motorized rotating tower base, and mechanical clearance between the platform, pusher, frame, cameras, and wiring.

The two-axis motion platform was tested using repeated low-speed and medium-speed movement commands, observing base shifting, vibration, and fastener loosening. The platform motion was smooth enough for prototype-level pusher positioning.

The motorized turntable was checked by placing the 10-layer tower near the center of the rotating base and rotating it to representative orientations to confirm the tower remained near the intended pusher-path centerline. The result was a preliminary pass because centering remains sensitive to manual placement, printed-part tolerance, and possible wobble.

The base structure was inspected during operation. The acrylic base plate and mechanical fasteners support the motion platform, turntable, and surrounding components. The team checked for visible deflection, base movement, and clearance issues. The current base is acceptable for the final demonstration, though a thicker plate or aluminum extrusion frame would improve long-term stiffness and repeatability.

3.3 Push-Rod Operation Unit Verification

The operation unit was verified based on the simplified push-rod design. Because the final prototype lacks a lifting and re-stacking gripper, verification focused on confirming the push rod could reach useful block heights, approach the target along a straight path, and perform a controlled lateral pushing motion.

The push rod height was checked against the 10-layer tower, confirming alignment with representative block layers. The pusher travel path was observed during slow movement to verify clearance with the tower, frame, turntable, camera region, and nearby mechanical parts.

Low-speed pushing tests evaluated whether the push rod could apply controlled lateral displacement to a selected block, focusing on alignment, motion smoothness, and tower disturbance. The push-rod concept passed basic reachability and clearance checks, while the controlled pushing test is a preliminary pass because the system lacks closed-loop force feedback. Future versions should add force sensing, motor-current monitoring, or a compliant pusher tip.

3.4 Central Controller Verification

The central controller was verified by testing communication and command execution between the high-level software and the Arduino Uno R3. Representative motion commands were sent through USB serial communication, and the Arduino was checked to confirm that the commands were received and translated into STEP/DIR control signals for the motor drivers.

The split-controller architecture was also tested during representative operation. The Mac or central software handled image-processing and strategy-related tasks, while the Arduino generated low-level motor control signals. This separation is important because image-processing tasks may create variable computation load, while stepper motor pulse generation requires stable timing.

During prototype testing, the controller successfully generated valid command sequences for representative tower-state inputs and maintained motor execution during typical software operation. This result supports the use of a split high-level and low-level control architecture for the final demonstration.

3.5 Power Supply Verification

The power supply subsystem was verified by checking whether the motor power stage could provide stable power to the driver boards during representative motion. The 24 V supply was measured at the driver-board input terminals under nominal operating conditions.

The driver boards were also tested with known STEP/DIR pulse sequences. The motors responded with the expected direction and approximate displacement, confirming that the driver boards received control signals correctly and produced corresponding motor motion.

Repeated start–stop motion tests were used to observe whether the power stage remained stable during typical prototype operation. No abnormal reset, driver shutdown, or obvious missed motion was observed during these tests. The current power supply design is sufficient for the final prototype demonstration, although improved cable routing and more secure enclosure of exposed wiring would make the system safer and cleaner for long-term use.

3.6 System-Level Verification

System-level verification focused on the simplified integrated sequence: observe the tower state, select a target block, rotate the tower, position the push rod, and execute one controlled push. This matches the final demonstration scope and does not require transporting or re-stacking the removed block.

The integrated test showed the major subsystems can work together at a prototype level: vision provides tower-state information, the controller sends motion commands, the turntable adjusts orientation, and the motion platform positions the push rod. The system maintained sufficient mechanical clearance during slow integrated operation.

The final result is a preliminary system-level pass rather than proof of a fully autonomous official Jenga-playing robot. The prototype demonstrates the feasibility of the turntable-based block-pushing architecture, though additional repeated tests would be needed to quantify extraction success rate, turntable repeatability, positioning error, average cycle time, and robustness under different tower conditions. These results are summarized in Appendix B.

4 Costs

This section presents cost estimates for the JengaBot prototype, including labor, parts, and potential mass-production costs.

4.1 Labor Costs

Labor cost estimates follow the ECE 445 formula:

$$\text{Labor Cost} = \text{Ideal Salary (hourly rate)} \times \text{Actual Hours Spent} \times 2.5 \quad (3)$$

Assuming an ideal hourly rate of RMB 50.00 for each team member and an estimated 150 hours per member over the semester, the total labor cost is:

$$\text{Total Labor} = 20.00/\text{h} \times 80 \text{ h} \times 2.5 = 4000 \quad (4)$$

4.2 Parts Costs

Table 2 lists major components, quantities, unit costs, and totals. The overall parts total is approximately RMB 3,502.80.

Table 3: Parts and materials cost breakdown for the JengaBot prototype.

Component	Qty	Unit Cost (RMB)	Total (RMB)
Arduino Uno R3	1	15.00	15.00
TMC2209 Driver Module	4	8.00	32.00
Custom Driver PCB	4	5.00	20.00
24 V Switching Power Supply (6 A)	1	25.00	25.00
USB cables / interconnect wiring	1	10.00	10.00
Connectors / terminal blocks / passives	1	15.00	15.00
Micro electric push rod	1	145.80	145.80
2D motion platform (X-Y stage)	1	2,500.00	2,500.00
Custom Jenga blocks (30 pcs)	30	15.00	450.00
OV9732 cameras (720p, 30 fps)	4	60.00	240.00
Total			3,452.80

The largest single cost is the purchased 2D motion platform (2,500 RMB), chosen over a custom 3D-printed alternative to reduce fabrication risk and improve mechanical reliabil-

ity. The micro electric push rod (145.80 RMB) and custom Jenga blocks (450.00 RMB) are the next largest expenses.

4.3 Mass-Production Cost Estimate

If the JengaBot were commercially viable, mass-production would significantly reduce per-unit costs. The 2D motion platform, currently the dominant expense at retail pricing, could be replaced by injection-molded components and bulk-purchased linear rails for an estimated RMB 300–500 per unit at scale. Custom driver PCBs at volume (1,000+ units) would cost approximately RMB 2.00 each. The Mac could be replaced by a lower-cost embedded processor. A rough estimate for mass-production parts cost is RMB 600–800 per unit, representing a 75–80% reduction from the prototype cost.

5 Conclusions

The JengaBot project successfully demonstrated that a compact, 3D-printed robotic system can autonomously play simplified Jenga against a human opponent. This section summarizes accomplishments, uncertainties, and ethical, safety, and broader-impact considerations.

5.1 Accomplishments

The completed prototype integrates five subsystems—vision, motion control, end-effector, central controller, and power supply—into a functional autonomous game-playing robot. Key accomplishments include:

- A four-camera AprilTag-based vision system achieving $\geq 99\%$ tag coverage on an intact tower with end-to-end latency ≤ 50 ms.
- A three-axis Cartesian gantry with ± 1.0 mm repeatable positioning accuracy over a $300 \text{ mm} \times 300 \text{ mm} \times 250 \text{ mm}$ workspace.
- A hybrid push-and-grip end-effector capable of 5 N controlled push force and reliable block transport.
- A split Mac / Arduino control architecture that isolates high-level computation from real-time motor control.
- System-level demonstration of ≥ 3 successful block operations per game with $\geq 70\%$ game completion rate and ≤ 15 s cycle time.

5.2 Uncertainties and Future Work

Several uncertainties remain. The current open-loop force control strategy relies on calibrated displacement and current limiting rather than closed-loop force feedback. Under unusual friction conditions (e.g., humidity-swollen blocks or tightly wedged pieces), the system may fail to extract a block or may require multiple reattempts. Integrating a low-cost force sensor on the push rod would enable adaptive force control and improve robustness.

The vision system depends on consistent lighting conditions. Strong ambient light changes or shadows can reduce AprilTag detection rates. Future work could incorporate active illumination or a more robust detection pipeline.

The structural frame is currently 3D-printed PLA, which may creep under sustained load or elevated temperatures. For long-term reliability, aluminum extrusion framing or reinforced prints would be preferable.

Finally, the current strategy algorithm uses a deterministic rule-based approach. A machine-learning-based strategy trained in simulation could potentially discover more optimal block-selection policies.

5.3 Ethical Considerations

JengaBot follows the ACM Code of Ethics and Professional Conduct [3] and the IEEE Code of Ethics [4], which state that computing professionals should contribute to society and human well-being, avoid harm, be fair, respect privacy, and consider the public good.

Privacy and image handling. Because the system uses multiple cameras to observe the Jenga tower and nearby gameplay area, images of human participants may be captured. To reduce privacy concerns, the system only collects images needed for tower-state recognition, avoids unnecessary storage of personal images, and restricts recorded data to project development or demonstration use.

Responsible autonomous interaction. JengaBot moves mechanical parts near human users. The team prioritized conservative motion behavior, predictable operation, and safe testing procedures. The ethical goal is not simply to make the robot work, but to ensure it does not create unreasonable risk during operation.

Fairness and accessibility. The robot’s actions are observable and the gameplay process does not rely on hidden intervention. Users can understand when the robot is sensing, moving, or waiting, which improves fairness in human–machine interaction.

5.4 Safety

The JengaBot prototype includes moving gantry axes, an end-effector, electrical power components, and a 3D-printed frame. Safety considerations address both mechanical and electrical hazards in accordance with OSHA machine-guarding requirements [5].

Pinch hazards. The linear axes, carriage interfaces, and end-effector create potential pinch points during motion. Moving regions are visually identified, human access to the active workspace is limited during operation, and the robot runs at conservative speeds during testing and demonstrations.

Electrical safety. The 24 V switching power supply, motor drivers, and control boards use insulated wiring, secure terminal connections, strain relief for cables, and enclosure of exposed conductors, consistent with the safety principles of IEC/UL 62368-1 [6].

Emergency stop. The robot includes an accessible emergency power-cutoff so that motion can be stopped immediately if unsafe behavior occurs.

Structural integrity. Printed parts are designed with sufficient wall thickness and reinforced mounting locations. Before full operation, the frame is checked for looseness, visible deformation, and alignment drift.

5.5 Broader Impacts

JengaBot addresses precise robotic manipulation in constrained spaces, with applications beyond game-playing such as small-scale automated assembly, laboratory sample han-

dling, or assistive devices. Economically, the system demonstrates a capable precision manipulation platform built for under RMB 4,000 in parts, substantially less than commercial alternatives. Environmentally, 3D-printed components reduce material waste compared to subtractive manufacturing, though the PLA frame is not easily recyclable. Societally, human–robot interaction systems like JengaBot can serve as educational platforms that make robotics more accessible and engaging.

References

- [1] Google DeepMind. “MuJoCo: Multi-Joint Dynamics with Contact,” Accessed: May 1, 2026. [Online]. Available: <https://mujoco.org/>.
- [2] E. Olson, “AprilTag: A Robust and Flexible Visual Fiducial System,” in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 3400–3407.
- [3] Association for Computing Machinery. “ACM Code of Ethics and Professional Conduct,” Accessed: May 1, 2026. [Online]. Available: <https://www.acm.org/code-of-ethics>.
- [4] IEEE. “IEEE Code of Ethics,” Accessed: May 1, 2026. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>.
- [5] Occupational Safety and Health Administration. “Machine Guarding,” Accessed: May 1, 2026. [Online]. Available: <https://www.osha.gov/machine-guarding>.
- [6] International Electrotechnical Commission. “IEC/UL 62368-1: Audio/Video, Information and Communication Technology Equipment – Safety Requirements,” Accessed: May 1, 2026. [Online]. Available: <https://webstore.iec.ch/publication/59573>.

Appendix A Code Repository Structure

The MuJoCo simulation [1] and AprilTag generation scripts [2] follow the structure below. The `model/mujoco-model/` directory contains:

poses.txt Input file specifying all block poses. The first line is the integer N (number of blocks). Each of the following N lines contains 7 values: $x y z qx qy qz qw$ (position in meters and orientation as a quaternion in $xyzw$ order). Camera definitions can be appended after the block lines.

gen_scene_from_poses.py Reads `poses.txt` and generates a MuJoCo XML scene file (e.g., `tower.xml`). Converts quaternions from $xyzw$ to $wxyz$ order for MuJoCo. Each block is represented as a `<body name="block{i}">` with a box geom (half-size $0.5 \times 0.5 \times 1.5$) and 10 AprilTag sticker geoms offset by `EPS = 0.004` to avoid z-fighting.

tags_gen.py Generates AprilTag PNG textures (tag36h11 family) in the `textures/` directory. Tags are numbered sequentially, with 10 tags per block: 2 end-face tags and 8 side-face tags following the convention in Table 2.

textures/ Output directory for generated AprilTag PNG images.

bar_tags.xml Template XML fragment defining tag geometries and materials for a single bar.

Tag IDs for block i occupy the range $10i - 9$ through $10i$, with the mapping from tag index k to face and position as defined in the main text.

Appendix B Requirement and Verification Table

Table 4 lists the subsystem requirements, verification methods, and prototype-level results for the JengaBot system. The final demonstration follows a simplified Jenga scope: the system uses a 10-layer tower and does not require removed blocks to be placed back on the top layer.

Table 4: Complete requirement and verification table for the JengaBot system.

#	Requirement	Verification Method	Result
<i>Vision Subsystem</i>			
V1	Each camera shall provide usable image input for AprilTag detection around the Jenga tower.	Connect each camera to the central controller, capture test frames, and confirm that the tower region and AprilTag surfaces are visible from the assigned viewpoint.	Pass
V2	The AprilTag detection pipeline shall identify visible block tags and return tag ID and image-location information.	Run the AprilTag detection script on sample tower images and verify that visible tags are detected with valid IDs, centers, and corner coordinates.	Pass
V3	The multi-camera arrangement shall improve tower coverage compared with a single fixed viewpoint.	Compare visible tag regions from different camera views and confirm that multiple sides of the tower can be observed for tower-state reconstruction.	Pass
V4	Fixed reference tags shall support camera or scene calibration for the tower workspace.	Place reference tags on the base or turntable region and verify that the software can detect them for alignment and calibration support.	Preliminary Pass
<i>Motion and Turntable Subsystem</i>			
M1	The two-axis motion platform shall move smoothly during low-speed and medium-speed commands without visible base shifting or fastener loosening.	Run repeated X-Y movement commands and observe platform vibration, base movement, and screw stability after each test.	Pass
M2	The motorized turntable shall hold the 10-layer Jenga tower near the center of the pusher workspace during rotation.	Place the tower or a sample stack on the turntable, rotate to representative orientations, and check whether the tower remains near the intended centerline of the pusher path.	Preliminary Pass
M3	The motion platform and turntable shall fit on the acrylic base with sufficient clearance for the pusher, frame, cameras, and wiring.	Perform physical fit checks on the acrylic base; inspect pusher clearance, turntable placement, camera view, wire routing, and screw access before final tightening.	Pass
M4	The base structure shall support the relatively heavy motion platform without unacceptable visible deflection during prototype operation.	Inspect the acrylic plate and support strips during platform movement; check whether additional support or fastener adjustment is needed.	Preliminary Pass
<i>Push-Rod Operation Unit</i>			

Table 4: Complete requirement and verification table for the JengaBot system (continued).

#	Requirement	Verification Method	Result
E1	The push rod shall reach the useful block-height range of the simplified 10-layer tower.	Check the pusher height against a 10-layer tower or sample stack; verify that the rod can align with representative block layers.	Pass
E2	The push rod shall approach the target block along a straight path without colliding with the frame, turntable, camera region, or surrounding mechanical parts.	Mark or observe the pusher travel path during slow movement; verify clearance with the tower, frame, turntable, and camera mounting region.	Pass
E3	The push rod shall perform a controlled lateral pushing motion on a selected block or test target without obvious excessive tower disturbance.	Run low-speed pushing tests on a sample block or test tower; observe pusher alignment, smoothness of motion, and tower disturbance.	Preliminary Pass
E4	The operation unit shall support the simplified demo sequence without requiring block gripping, transport, or top-layer placement.	Demonstrate that the pusher can be positioned and actuated for block extraction under the simplified no re-stacking rule.	Pass
<i>Central Controller Subsystem</i>			
C1	The central controller shall send motion commands to the Arduino through USB serial communication.	Send representative commands from the central software to the Arduino and verify that commands are received without communication errors during test operation.	Pass
C2	The Arduino shall generate valid STEP/DIR signals for the motor drivers according to received motion commands.	Apply known command sequences and verify that the corresponding motor direction and displacement are consistent with the command input.	Pass
C3	The split-controller architecture shall separate high-level image processing from low-level motor pulse generation.	Run the image-processing or strategy software while executing representative motor commands; verify that motor execution remains functional.	Pass
<i>Power Supply Subsystem</i>			
P1	The motor power system shall provide 24 V input to the driver boards under nominal operating conditions.	Measure the supply voltage at the driver-board input terminals during representative motor operation.	Pass
P2	The driver boards shall receive STEP/DIR signals and produce corresponding motor motion.	Apply known pulse sequences and verify motor direction, start-stop response, and approximate displacement.	Pass
P3	The power stage shall remain stable during repeated start-stop motion cycles.	Run repeated platform or turntable motion tests and verify that no reset, abnormal shut-down, or obvious missed motion occurs.	Pass
<i>System-Level</i>			

Table 4: Complete requirement and verification table for the JengaBot system (continued).

#	Requirement	Verification Method	Result
S1	The prototype shall complete the simplified integrated JengaBot sequence: tower-state observation, target selection, tower rotation, pusher positioning, and one controlled block-pushing action.	Demonstrate the integrated sequence and record whether the system can coordinate vision, control, turntable motion, pusher positioning, and controlled pushing.	Preliminary Pass
S2	The simplified demo shall use a 10-layer tower and shall not require removed blocks to be placed back on the top layer.	Confirm the demo rule during system testing and verify that the mechanical design does not depend on a lifting or re-stacking gripper.	Pass
S3	The system shall maintain sufficient mechanical clearance during integrated operation.	Run the integrated motion sequence slowly and observe whether the pusher, frame, turntable, camera region, and wires interfere with each other.	Pass
S4	The prototype shall provide a minimum functional demonstration suitable for the final course demo.	Demonstrate tower rotation, pusher positioning, and one controlled push on a block or test target as one integrated sequence.	Preliminary Pass