

VOICE-CONTROLLED ROBOTIC STUDY ASSISTANT

By

Yicheng Chen (yc69)

Shuohan Fang (shuohan5)

Jiaxuan He (he59)

Qi Jin (qjin6)

Final Report for ECE 445, Senior Design, Spring 2026

Advisor: Hua Chen

TA: Xihe Shao

15 May 2026

Team No. 46

Abstract

This report documents the design and verification of a voice-controlled robotic study assistant for users with upper-limb motor impairments. The prototype accepts offline voice commands and turns the pages of a book on a desktop workstation. The team replaced the vacuum-suction concept of the design document with a silicone friction wheel: driven by an N20 gear motor and two NEMA 17 stepper axes, it rolls the top page upward, after which a two-servo sweep bar carries the page across the binding while two paperweights hold the remaining pages flat. A Raspberry Pi 5 runs the control software; the Vosk engine recognizes commands offline, and a read command captures the page with a USB camera and sends it to a PaddleOCR service whose text is spoken by espeak-ng. The prototype demonstrated end-to-end page turning and the voice-to-speech pipeline; several quantitative metrics were being characterized at the time of writing.

Contents

- 1. Introduction..... 1
 - 1.1 Problem Statement 1
 - 1.2 Solution Overview..... 1
 - 1.3 High-Level Requirements 2
- 2. Design 3
 - 2.1 System Architecture 3
 - 2.2 Robotic Page-Turning Mechanism..... 4
 - 2.3 Document Workstation 5
 - 2.4 Power and Electronics..... 6
 - 2.5 Control Software 7
 - 2.6 Vision and OCR Pipeline 8
 - 2.7 Voice Command and Audio Output..... 9
- 3. Verification 10
 - 3.1 Page-Turning Mechanism 10
 - 3.2 Document Workstation 10
 - 3.3 Power and Electronics..... 10
 - 3.4 Vision and OCR Pipeline 10
 - 3.5 Voice Command and Audio Output..... 11
- 4. Cost and Schedule..... 12
 - 4.1 Parts..... 12
 - 4.2 Labor 12
 - 4.3 Schedule 13
- 5. Conclusion 14
 - 5.1 Accomplishments 14
 - 5.2 Uncertainties..... 14
 - 5.3 Ethical Considerations..... 15
 - 5.4 Future Work 15
- References..... 17
- Appendix A Requirement and Verification Tables 18

Appendix B Additional Photographs..... 20

1. Introduction

This chapter introduces the motivation for the project, states the accessibility problem it addresses, summarizes the as-built solution, and lists the high-level requirements against which the prototype is evaluated in Chapter 3.

1.1 Problem Statement

Users with upper-limb mobility limitations encounter substantial physical barriers when interacting with printed materials such as bound textbooks, stapled lecture notes, and printed manuals. Basic actions such as turning a page, holding a document flat, or moving between sections require continuous physical effort or the help of another person. Although digital accessibility tools are widely available, a large fraction of academic and professional content still exists only in physical form, often with handwritten annotations or non-standard formatting that is awkward to digitize. An assistive device is therefore needed that lets a seated user manipulate printed documents hands-free, with predictable mechanical behavior and audible content playback.

1.2 Solution Overview

The team built a desktop-scale robotic reading assistant that turns the pages of a physical book in response to spoken commands and reads the current page aloud. The user issues one of four short commands—"forward," "backward," "read," or "query"—which a Raspberry Pi 5 maps to a mechanical page-turn or to a vision-and-speech action. Page turning is performed by a silicone friction wheel that rolls the top sheet upward, followed by a two-servo gimbal sweep bar that carries the lifted page across the binding; two servo-actuated paperweights hold the remaining pages flat. The complete assembly is shown in Figure 1.1.

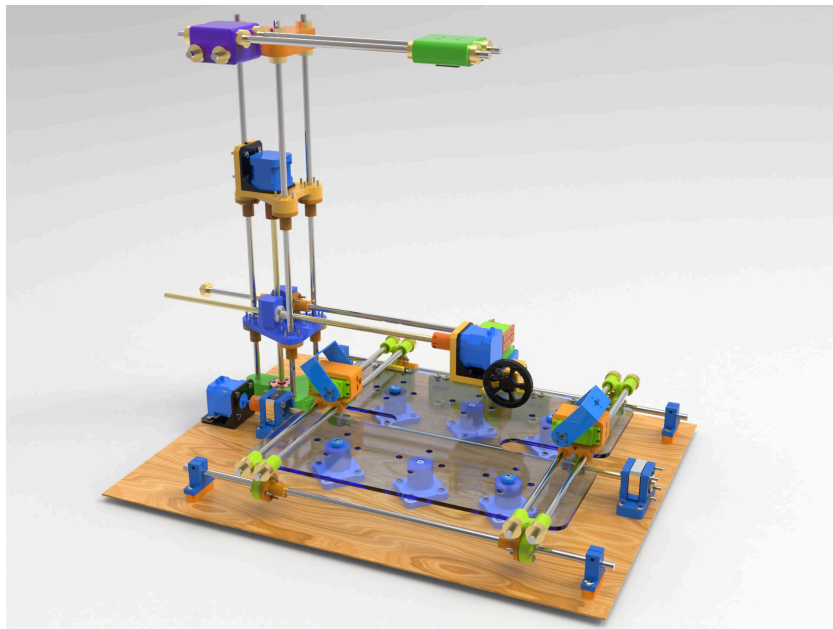


Figure 1.1 Computer-aided design model of the assembled study assistant.

For the "read" command an overhead USB camera captures the open page, a PaddleOCR service extracts the printed text, and the espeak-ng engine synthesizes speech. The "query" command additionally sends the extracted text to a large language model so the system can summarize or explain the page rather than only read it verbatim. The design evolved during the semester: the vacuum-suction page lifter, discrete power tree, and hardware voice module described in the design document were each replaced with simpler alternatives, as discussed in Chapter 2.

1.3 High-Level Requirements

The prototype was evaluated against four high-level requirements carried forward from the project proposal:

1. Voice control and latency: the system shall recognize its predefined spoken commands with at least 80 % accuracy and initiate the corresponding action within 3 s of the end of speech.
2. Document handling: the workstation shall hold a bound book flat and accommodate common academic page sizes from ISO A5 to ISO A4, keeping the page surface stable enough for overhead imaging.
3. Page turning: the mechanism shall turn a single page forward or backward within 4 s per cycle and shall achieve at least 80 % success over 20 continuous trials without tearing or permanently creasing the paper.
4. Vision, recognition, and audio output: the system shall capture the open page, extract its text with at least 90 % character accuracy, and play the synthesized speech back to the user.

2. Design

This chapter describes the as-built design of the prototype. Section 2.1 presents the overall system architecture. Sections 2.2 through 2.7 describe the six subsystems in turn—the robotic page-turning mechanism, the document workstation, the power and electronics, the control software, the vision and optical character recognition (OCR) pipeline, and the voice command and audio output. Each section also records the principal design alternatives that were considered and the changes made relative to the design document.

2.1 System Architecture

The prototype is organized as six interacting subsystems. The robotic page-turning mechanism performs the physical page manipulation. The document workstation supports the book and holds the unturned pages flat. The power and electronics subsystem distributes energy and hosts the motor and servo drivers. The control software, running on a Raspberry Pi 5, sequences all actions through a state machine. The vision and OCR pipeline captures and reads the open page, and the voice command and audio output subsystem provides the spoken user interface. Figure 2.1 shows the high-level signal and data flow between these subsystems.

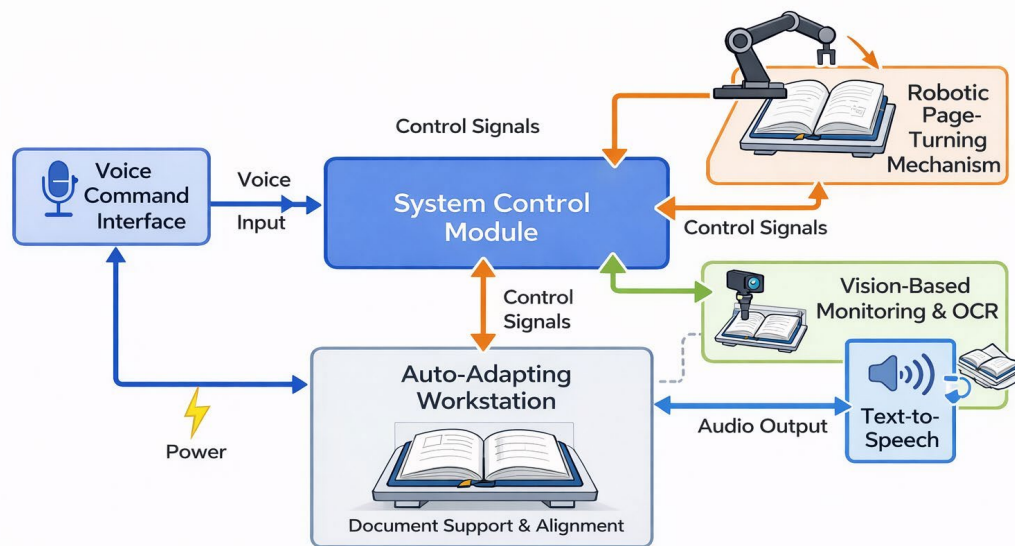


Figure 2.1 System block diagram of the voice-controlled robotic study assistant.

Operation is centralized on the Raspberry Pi 5. A spoken command is recognized locally and dispatched by the state machine. Navigation commands drive the page-turning actuators through the expansion board; the "read" and "query" commands trigger an image capture, an OCR request, and speech synthesis. The workstation passively supports the book throughout, and the power subsystem feeds a regulated logic supply to the Raspberry Pi and a separate 12 V supply to the motors and servos.

2.2 Robotic Page-Turning Mechanism

The page-turning mechanism replaces the vacuum-suction concept of the design document with a silicone friction-wheel approach. A 60 mm diameter, 9 mm wide silicone friction wheel is driven by a 12 V N20 gear motor. To turn a page, the wheel is lowered onto the top sheet and rotated; friction between the silicone surface and the paper rolls the free edge of the page upward into a standing curl. Figure 2.2 shows the friction-wheel head as built.

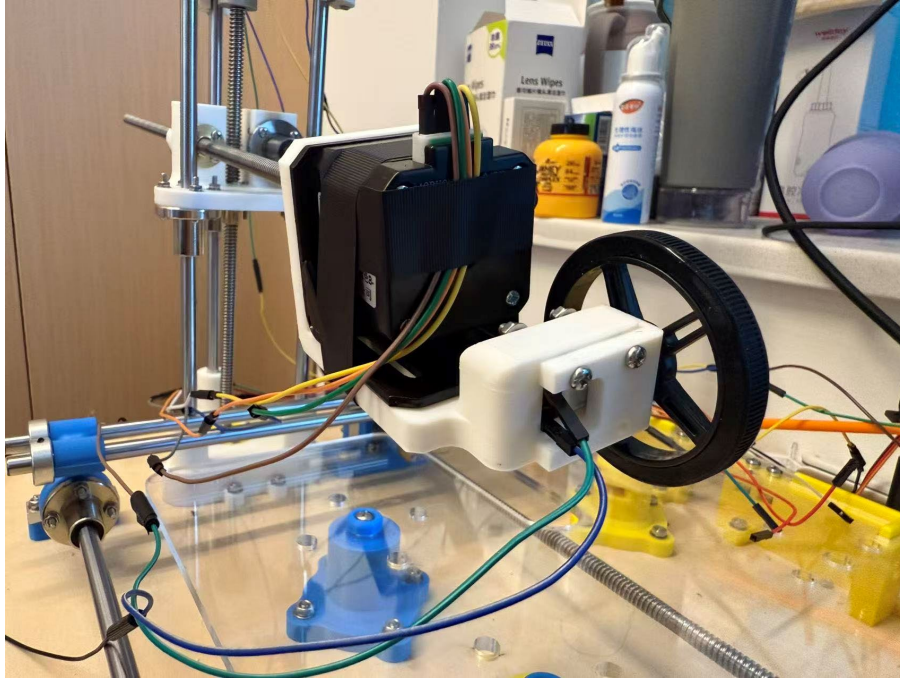


Figure 2.2 Friction-wheel page-turning head, showing the silicone wheel, its drive motor, and the two-axis motion stage.

The friction wheel is carried on a two-axis motion stage. A horizontal axis, built from a T8 lead screw and an 8 mm guide rod driven by a NEMA 17 stepper motor, positions the wheel across the page over a travel of about 350 mm; a vertical axis driven by a second NEMA 17 stepper [1] raises and lowers the wheel over a travel of about 180 mm. Locating the wheel by stepper position lets the same mechanism reach books of different sizes and thicknesses.

Successful single-page pickup depends on the friction force at the wheel exceeding the resistance that holds the page down. The governing condition is given by Equation (2.1), in which the friction force equals the static friction coefficient between the silicone wheel and the paper multiplied by the contact normal force, and the resistance term combines the inter-page friction and the binding stiffness:

$$F_{friction} = \mu_s N > F_{resist} \quad (2.1)$$

Once the free edge of the page has been curled upward, a sweep bar completes the turn. The sweep bar is a two-axis gimbal built from two MG996R metal-gear servos in a pan-and-tilt arrangement,

with a sweep arm whose tip lies about 108 mm from the upper servo axis. The lower servo rotates the arm across the binding while the upper servo sets the arm height, so that the arm slides under the lifted page and carries it to the opposite side. Figure 2.3 shows the gimbal.

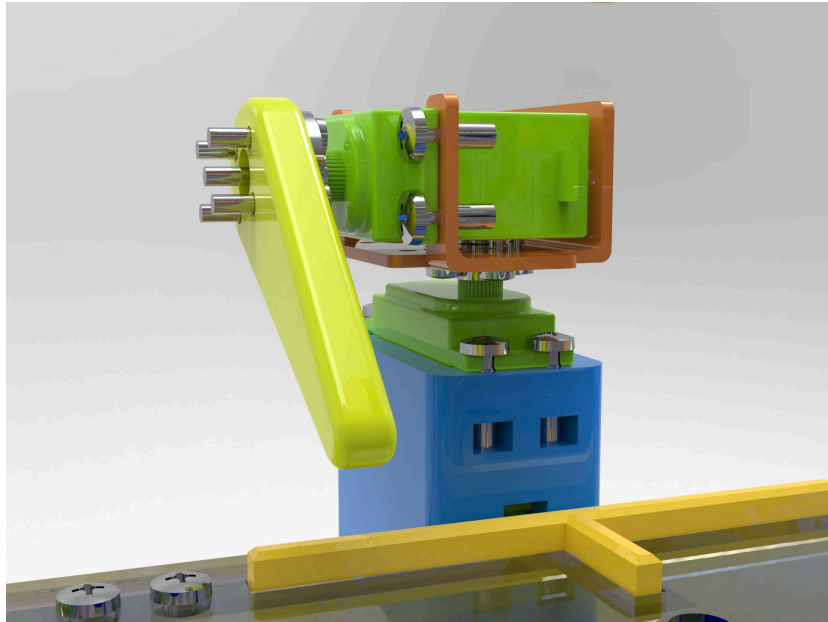


Figure 2.3 Two-servo gimbal sweep bar that carries a lifted page across the binding.

Two design alternatives were considered before the friction-wheel mechanism was adopted. A vacuum suction cup, the approach taken in the design document, was set aside because it requires an air pump and tubing that raise cost and complicate a portable demonstration, and because the suction force is sensitive to supply voltage and paper condition. A printer-style feed roller was also considered but rejected, as a roller small enough for the assembly risked the frame contacting the page before the roller, and reliable sizing data were not available. The friction wheel needs no air path, is inexpensive, and tolerates a range of paper types.

One mechanical change was made during prototyping. The sweep arm was initially about 58 mm long; bench testing showed that this was too short to lift and clear the page reliably, so the arm was lengthened to about 108 mm, which corrected the problem.

2.3 Document Workstation

The document workstation supports the book and holds the unturned pages flat so that the friction wheel and camera see a stable surface. The frame is built on a wooden baseboard with an overall footprint of about 600 mm by 450 mm and a height of about 500 mm. Structural members are 8 mm steel smooth rods located by locking collars and SK8 shaft supports, with LMF8UU linear bearings carrying the moving stages. A wooden baseboard was chosen over 2020 aluminum extrusion because it is easy to machine, can be re-drilled as the layout is refined, and is inexpensive; aluminum extrusion would have required external machining and is harder to adjust once cut.

Two paperweights hold the left and right page stacks flat. Each paperweight is a 3D-printed arm driven by an MG996R metal-gear servo, which lowers the arm onto the margin of the book and applies a holding force sufficient to keep the pages flat without abrading them. Figure 2.4 shows one paperweight arm as built. MG996R servos were chosen over stepper motors or solenoids because they are compact, need few supporting parts, and provide adequate torque with simple position control.

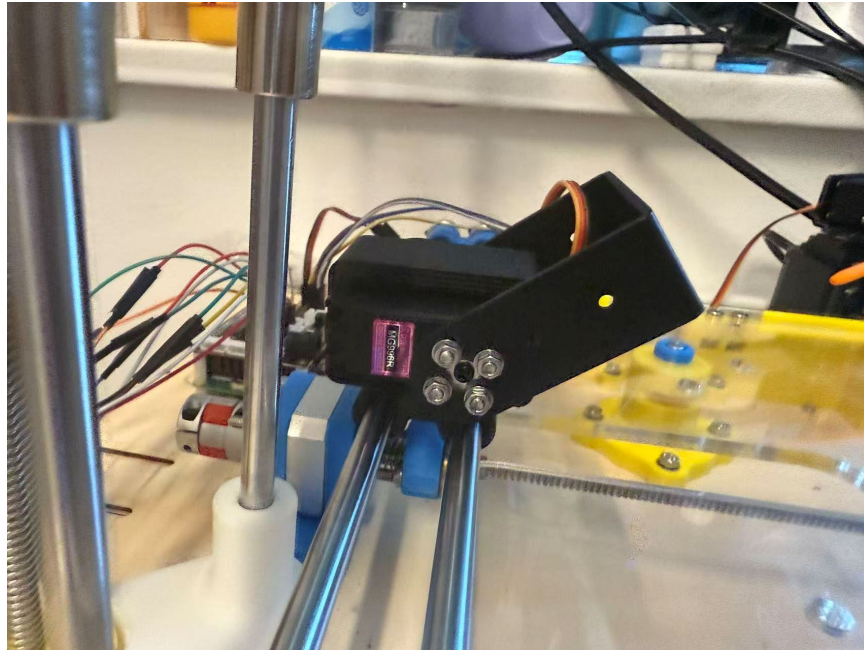


Figure 2.4 Servo-actuated paperweight arm that holds a page stack flat.

The design document specified a third stepper motor to clamp the book automatically as part of an auto-adapting interface. Because the Raspberry Pi expansion board did not have enough free motor channels for a third stepper, this clamping motion was changed to a manual adjustment: the operator sets the book position once before a session, after which the paperweights and friction wheel operate automatically. The workstation still accommodates the ISO A5 to ISO A4 page range [2], but the initial book placement is no longer motorized.

2.4 Power and Electronics

The power and electronics subsystem was simplified considerably relative to the design document. The Raspberry Pi 5 [3] is powered from its official USB-C supply, and the 3.3 V logic rail is provided by the expansion board, so the discrete LM2596 buck converter and 3.3 V low-dropout regulator of the design document were not built. A separate 12 V, 5 A adapter supplies the motors and servos.

All motor and servo drivers are integrated on a Raspberry Pi servo and motor expansion board that mounts directly on the Pi header, so no separate stepper-driver boards and no custom printed circuit board were fabricated. The board carries the connections to the two NEMA 17 steppers, the N20

gear motor, and the four MG996R servos. Figure 2.5 shows the Raspberry Pi 5 and the expansion board as wired. Over-current protection is provided by the expansion board and the adapter rather than by a discrete fuse, and a power switch on the board lets the operator cut motor power.

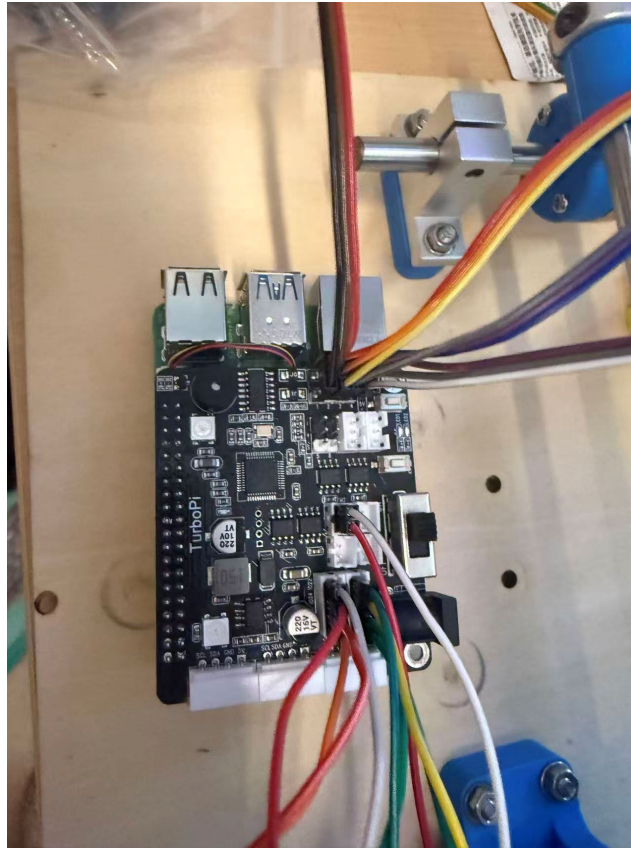


Figure 2.5 Raspberry Pi 5 and the servo and motor expansion board as wired in the prototype.

The integrated expansion board was chosen over a custom PCB carrying discrete LM2596 and stepper-driver circuits. During prototyping the wiring changed frequently as the mechanism was tuned, which made a fixed PCB layout impractical within the project schedule; the expansion board allowed the electronics to be rewired quickly with jumper leads.

2.5 Control Software

All high-level behavior runs on the Raspberry Pi 5. A state machine begins in an idle state and waits for a recognized voice command. A "forward" or "backward" command moves the system into a page-turn state that sequences the paperweights, the friction-wheel motion stages, the friction wheel, and the sweep-bar gimbal, then returns to idle. A "read" command moves the system into a capture-and-OCR state followed by a speech-output state; a "query" command follows the same path but routes the recognized text through a large language model before speech synthesis. The top-level control flow is shown in Figure 2.6.

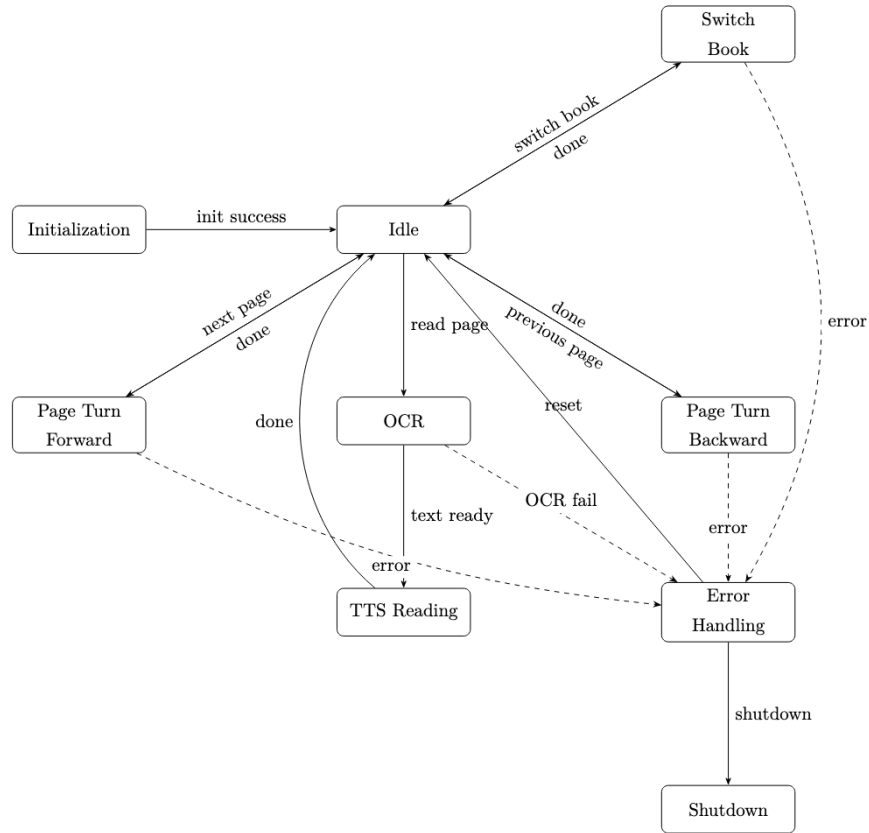


Figure 2.6 Top-level control state machine.

The controller drives the actuators in open loop: each motion uses fixed, pre-tuned travel and timing parameters, and the system does not yet sense whether the friction wheel has actually contacted or separated a page. This keeps the control software simple but, as discussed in Chapter 5, makes page-turn reliability sensitive to book type and thickness. The "query" state, which summarizes a page through a language model, was added beyond the scope of the design document; the original "switch book" and "shutdown" commands were dropped to keep the command set small and reliably distinguishable.

2.6 Vision and OCR Pipeline

The vision and OCR pipeline reads the open page on demand. On a "read" or "query" command an overhead USB camera captures the current page at 1080p resolution through OpenCV. Because the Raspberry Pi 5 used in the prototype has only 1 GB of memory, optical character recognition is not performed on the Pi itself. Instead the captured image is sent over the local network as an HTTP POST request to a PaddleOCR service [4].

The OCR service runs PaddleOCR 3.5.0 inside a FastAPI application [5] on a laptop, listening on port 8000. It receives the page image, runs text detection and recognition, and returns the extracted text to the Raspberry Pi. PaddleOCR was chosen over its lightweight model variant and over running OCR locally on the Pi: the lightweight model did not give acceptable accuracy, while the

full model exceeded the memory available on the 1 GB Pi. Offloading the full model to a networked service preserved recognition quality at the cost of a dependence on the laptop and the local network, a trade-off revisited in Chapter 5.

2.7 Voice Command and Audio Output

Voice recognition runs entirely in software on the Raspberry Pi using the Vosk offline speech recognition toolkit [6]. The design document's hardware Voice Recognition V3 module and its UART link were not used; the Vosk-based approach removes the dedicated hardware module and keeps all recognition on the main processor. A small command vocabulary—"forward," "backward," "read," and "query"—keeps the recognition task well-conditioned and unambiguous. Vosk was selected over cloud speech services and over heavier offline models such as Whisper because the command set is small and distinct, so a light offline recognizer is sufficient and avoids any network dependence for the voice interface.

When the recognized command is "read," the text returned by the OCR service is passed to the `espeak-ng` engine [7], which synthesizes Chinese speech at a 22.05 kHz sample rate. When the command is "query," the OCR text is first sent to the Qwen large language model [8], which returns a short explanation of the page that is then spoken through `espeak-ng`. Audio is produced by `espeak-ng` directly; the powered speaker for the prototype had not been delivered at the time of writing, so the audio path was exercised on the development machine.

3. Verification

This chapter reports the verification status of each subsystem as of the time of writing. Quantitative results that had been measured are stated directly; metrics that were confirmed only qualitatively, or that were scheduled for measurement at the May 18 demonstration, are identified as such. Tables A.1 through A.5 in Appendix A give the corresponding requirement and verification details for the five subsystems.

3.1 Page-Turning Mechanism

The page-turning mechanism was exercised with real books during integration. Qualitative testing confirmed that the friction wheel curls the free edge of a page upward and that the sweep-bar gimbal carries the lifted page across the binding; no tearing or permanent creasing was observed in these trials. In preliminary continuous trials the page-turn operation succeeded in roughly half of the attempts, with success strongly dependent on book type and page thickness. The team attributes the shortfall to open-loop actuation and the absence of contact feedback, and was refining the friction-wheel timing and the coordination with the paperweights ahead of the final demonstration. The formal 20-trial success rate and the per-cycle turn time were to be measured at the demonstration.

3.2 Document Workstation

The document workstation was verified qualitatively. The two servo-actuated paperweights were observed to hold the left and right page stacks flat across the tested range of books, presenting a stable surface to the friction wheel and the camera. The normal holding force applied by the paperweights was not measured quantitatively; the force was confirmed only by observation to be sufficient to keep the pages flat without visibly damaging them.

3.3 Power and Electronics

The power and electronics subsystem was verified during normal operation. The Raspberry Pi supply voltage remained stable while the system ran, and the 12 V motor supply showed no observable voltage anomaly when the steppers, the gear motor, and the servos were active together. Quantitative measurements of supply ripple and of the peak current drawn by the motor rail were not taken; the subsystem was judged acceptable from its stable behavior under the full operating load.

3.4 Vision and OCR Pipeline

The vision and OCR pipeline was validated on the development machine. Page images captured by the USB camera were sent to the PaddleOCR service, which returned recognized text that was then passed to speech synthesis; the full capture-to-text path ran correctly end to end. The character accuracy on a controlled ground-truth page and the image-capture-to-text latency were to be measured at the final demonstration, once the pipeline was deployed on the Raspberry Pi.

3.5 Voice Command and Audio Output

The voice command and audio output subsystem was exercised throughout development. The Vosk recognizer correctly identified the four command words in informal testing, and the recognized commands triggered the intended state transitions. The espeak-ng engine produced intelligible Chinese speech from both the OCR text and the language-model summary on the development machine. A formal recognition-accuracy count over repeated trials, the end-to-end command latency, and the played-back loudness were deferred until the powered speaker was installed and the software was deployed on the Raspberry Pi.

4. Cost and Schedule

This chapter summarizes the prototype cost and the project schedule. Section 4.1 lists the bill of materials, Section 4.2 estimates labor cost using the formula prescribed by the ECE 445 guidelines, and Section 4.3 records the work carried out each week by each team member.

4.1 Parts

Table 4.1 lists the parts purchased for the prototype. Prices are the actual amounts paid, taken from the project procurement records; the 12 V adapter price is an estimate. The total material cost of the prototype was approximately 1,325 RMB.

Table 4.1 Prototype bill of materials.

Part	Qty	Cost (RMB)	Source / note
Raspberry Pi 5 (1 GB) kit	1	399.00	Main controller.
Servo and motor expansion board	1	329.00	Integrated motor / servo drivers.
USB camera (1080p)	1	79.00	Overhead page imaging.
NEMA 17 (42 mm) stepper motors and accessories	2	70.80	Friction-wheel X and Z axes.
MG996R metal-gear servos	4	60.92	Sweep-bar gimbal (2) and paperweights (2).
N20 gear motor (GA12-N20, 12 V)	1	16.33	Drives the friction wheel.
Silicone friction wheel (60 mm)	1	5.00	Page-lifting wheel.
T8 lead screws and nuts	1 set	39.00	Horizontal motion axis.
8 mm smooth rods	1 set	51.80	Frame and guide rails.
Locking collars	1 set	88.43	Rod location.
LMF8UU linear bearings	1 set	30.60	Moving stages.
SK8 shaft supports	1 set	17.00	Rod end supports.
Shaft couplers	1 set	26.00	Motor-to-screw coupling.
Wooden baseboard	1	22.30	Frame base.
Screws and nuts (M3 / M4 assortment)	1 set	58.85	Fasteners.
12 V / 5 A power adapter	1	31.00 (est.)	Motor supply.
Total		1,325.03	Prototype material cost only.

4.2 Labor

Labor cost is estimated with the ECE 445 formula, ideal hourly rate multiplied by hours worked multiplied by 2.5. An hourly rate of 10 RMB is used, consistent with the design document. Table 4.2 gives the breakdown by team member. The 2.5 multiplier accounts for university facility overhead and equipment use.

Table 4.2 Estimated labor cost.

Team member	Role	Rate (RMB/h)	Hours	x2.5	Cost (RMB)
Jiaxuan He	Mechanical	10	60	2.5	1,500
Qi Jin	Electronics and integration	10	60	2.5	1,500
Yicheng Chen	Software and vision	10	30	2.5	750
Shuohan Fang	Documentation and verification	10	40	2.5	1,000
Total			190		4,750

Combining Table 4.1 and Table 4.2, the total estimated project cost is approximately 6,075 RMB: 1,325 RMB of materials and 4,750 RMB of labor. The material figure excludes shipping and shared laboratory tooling.

4.3 Schedule

Table 4.3 records the work actually carried out each week by each team member, from the start of detailed design through the week of the demonstration.

Table 4.3 Weekly work by team member.

Week	Jiaxuan He (Mechanical)	Qi Jin (Electronics)	Yicheng Chen (Software)	Shuohan Fang (Documentation)
4/6	Refined the mechanical layout and 3D models; modeled the friction-wheel motion stage and the baseboard hole layout.	Set up the Raspberry Pi environment; tested the servos.	Built the Vosk environment and command vocabulary; voice module working.	Cleaned the BOM spreadsheet; standardized the verification-record templates.
4/13	Prepared 3D-print and board-cutting files; gathered lead screws, rods, bearings, and brackets.	Tested the DC gear motor and the stepper motors.	Built the PaddleOCR service on a laptop and the Raspberry Pi HTTP client.	Finalized the procurement priority list and checked lead times.
4/20	Completed the first batch of printed and cut parts; assembled the main support structure.	Tested combined-motion programming of the actuators.	Implemented the book-controller logic and simulated the overall call flow.	Placed component orders; drafted the integration verification worksheets.
4/27	Assembled the friction wheel, the N20 motor, the sweep gimbal, and the two paperweights.	Ran page-turning tests with real books.	Refactored and optimized the book-controller code.	Received and inspected parts; updated the BOM and issue log.
5/4	Mechanical bring-up; tuned the friction wheel, sweep bar, and paperweight coordination.	Tested additional books; began camera trials.	Validated the end-to-end pipeline on a PC with recorded audio and images.	Set up the page-turn and OCR verification procedures.
5/11	Optimized the sweep, paperweight, and wheel coordination; revised the camera mounting.	Investigated the camera bottleneck; focused on page-turn stability.	Connected the live camera and microphone and ran the full pipeline in real time.	Ran the first full requirement-check round; compiled the rework list.
5/18	Final assembly, on-site tuning, and mechanical stability checks for the demonstration.	Final system integration and demonstration debugging.	Deployed the software stack to the Raspberry Pi and completed hardware integration.	Polished the report and prepared the demonstration script.

5. Conclusion

This chapter summarizes what the project accomplished, the uncertainties that remain, the ethical considerations that shaped the design, and recommended future work.

5.1 Accomplishments

The team built a working desktop prototype of a voice-controlled robotic study assistant. The mechanical assembly—the friction-wheel page-turning head, its two-axis motion stage, the two-servo sweep-bar gimbal, and the two paperweights—was completed and integrated on the Raspberry Pi 5 through the expansion board. The software pipeline that links offline voice recognition, page-image capture, networked OCR, speech synthesis, and the language-model query was validated end to end on the development machine. Page turning was demonstrated on suitable books, and the system reads a captured page aloud. The "query" command, which summarizes a page through a large language model, extends the original concept beyond simple verbatim reading.

The integrated demonstration on the physical prototype was scheduled for May 18, 2026.

5.2 Uncertainties

Several uncertainties remain. The most significant is page-turn reliability: in preliminary continuous trials the page-turn operation succeeded in roughly half of the attempts, and success depended strongly on book type and page thickness. The root cause is open-loop actuation—the mechanism uses fixed motion parameters and has no sensor to confirm that the friction wheel has contacted or separated a single page—so it cannot adapt to a page that does not lift as expected. Second, the OCR pipeline depends on a laptop service and the local network, because the 1 GB Raspberry Pi 5 cannot host the full PaddleOCR model; a network or laptop fault would disable page reading. Third, the powered speaker had not been delivered at the time of writing, so the audio output path was verified only on the development machine. Finally, several quantitative metrics—page-turn success rate over 20 trials, OCR character accuracy, command latency, and playback loudness—had not yet been measured and were to be characterized at the demonstration.

Relative to the 80 % target, the preliminary page-turning result shows that the mechanism is functional but not yet robust. A success rate near 50 % means that the prototype can demonstrate the intended physical sequence, but it cannot be considered reliable for unattended assistive use. This gap is important because the user experience depends more on repeated reliability than on a single successful page turn. A user with limited upper-limb mobility would not benefit from a device that requires frequent manual reset, even if individual successful trials look correct. Therefore, the main unresolved engineering problem is not whether the friction-wheel concept can lift a page, but whether the controller can make the pickup repeatable across different paper stiffnesses, book curvatures, and page positions.

5.3 Ethical Considerations

Developing an assistive device for users with motor impairments carries ethical obligations regarding accessibility, honesty, privacy, and safety. In line with the IEEE Code of Ethics [9], the team reports the prototype's capabilities without overstatement: page-turn reliability is currently limited and depends on book type, the OCR stage depends on an external service, and the audio output had not been fully verified on hardware. The device should therefore be regarded as a laboratory prototype rather than a finished product.

A project-specific concern is data privacy. The camera and microphone capture page content and speech. Voice recognition with Vosk and speech synthesis with espeak-ng run locally, and the OCR request is sent only to a service on the team's own local network. The "query" command, however, sends the recognized page text to an external large-language-model service, so a user choosing that command should be aware that page text leaves the device; only text, not images, is transmitted, and the query is made only on an explicit command.

On safety, the moving friction wheel, the sweep-bar gimbal, and the motion stages create pinch points. Consistent with OSHA standard 29 CFR 1910.212 on machine guarding [10], the mechanism should be operated with hands clear of the moving envelope. A power switch on the expansion board lets the operator cut motor power, but a dedicated latching emergency-stop button was not integrated and is recommended for future revisions.

5.4 Future Work

The final prototype demonstrated the main system workflow, but several parts of the design remain better suited for a controlled laboratory demonstration than for reliable assistive use. The future work should therefore focus less on adding new functions and more on improving repeatability, self-containment, and user safety. In particular, the page-turning mechanism needs feedback so that the controller can respond to failed pickups instead of executing a fixed open-loop sequence. The OCR pipeline should also be moved closer to the embedded system so that the reading function does not depend on an external laptop. Based on these limitations, the following improvements would most directly strengthen the next version of the device.

1. Add closed-loop feedback. A contact or page-detection sensor at the friction wheel would let the controller confirm a successful pickup and retry a failed one, which directly addresses the page-turn reliability shortfall.
2. Move OCR on-board. A Raspberry Pi with more memory, or a small accelerator, would allow the full PaddleOCR model to run locally and remove the dependence on the laptop service and the local network.
3. Improve mechanical robustness. Tuning the friction-wheel and paperweight geometry for a wider range of book thicknesses, and stiffening the frame, would reduce the sensitivity to book type.
4. Complete the hardware. Installing the powered speaker and a dedicated emergency-stop button would finish the audio path and improve operator safety.

Among these improvements, closed-loop page detection should be the first priority because it addresses the largest measured performance gap. A simple sensor near the friction wheel or sweep path would allow the controller to distinguish three cases: no page lifted, one page lifted, or multiple pages disturbed. The controller could then retry with a different wheel position or normal force instead of continuing blindly through the sweep sequence. After this reliability issue is improved, moving OCR onto the device and adding the speaker and emergency stop would make the system more self-contained and safer, but those changes would not by themselves solve the main page-turning limitation.

References

- [1] *NEMA 17 Stepper Motor Datasheet*, Pololu Corp., Las Vegas, NV, USA, 2021. [Online]. Available: <https://www.pololu.com/file/0J714/SY42STH38-1684A.pdf>
- [2] *Writing Paper and Certain Classes of Printed Matter—Trimmed Sizes—A and B Series*, ISO 216:2007, International Organization for Standardization, Geneva, Switzerland, 2007.
- [3] *Raspberry Pi 5 Product Brief*, Raspberry Pi Ltd., Cambridge, U.K., 2023. [Online]. Available: <https://www.raspberrypi.com/products/raspberry-pi-5/>
- [4] PaddlePaddle. *PaddleOCR*, version 3.5.0. (2026). [Online]. Available: <https://github.com/PaddlePaddle/PaddleOCR>
- [5] *FastAPI*, version 0.136.1. (2026). [Online]. Available: <https://github.com/fastapi/fastapi>
- [6] Alpha Cephei. *Vosk Speech Recognition Toolkit*, version 0.3.44. (2026). [Online]. Available: <https://github.com/alphacep/vosk-api>
- [7] *eSpeak NG Text-to-Speech*. (2026). [Online]. Available: <https://github.com/espeak-ng/espeak-ng>
- [8] Alibaba Cloud. *Qwen (Tongyi Qianwen) Large Language Model*. (2026). [Online]. Available: <https://www.aliyun.com/product/tongyi>
- [9] *IEEE Code of Ethics*, IEEE, Piscataway, NJ, USA, 2020. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>
- [10] *General Requirements for All Machines*, OSHA Standard 29 CFR 1910.212, Occupational Safety and Health Administration, Washington, DC, USA. Accessed: May 2026. [Online]. Available: <https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.212>

Appendix A Requirement and Verification Tables

Tables A.1 through A.5 list the requirement and verification details for the five subsystems. The result column reflects the verification status at the time of writing; entries marked "pending" were to be measured at the May 18 demonstration.

Table A.1 Page-turning mechanism.

Requirement	Verification procedure	Result
Turn a single page forward or backward within 4 s per cycle.	Issue a page-turn command on a bound book and time the cycle from motion start to the arm returning to rest.	Pending.
Achieve at least 80 % success over 20 continuous page-turn trials with no torn or permanently creased pages.	Run 20 continuous forward and backward turns; inspect each page for damage and record success or failure.	Preliminary trials: about 50 % success, book-type dependent; no page damage observed. Formal 20-trial count pending, with each trial classified as a clean single-page turn, missed turn, multi-page pickup, or damage event.

Table A.2 Document workstation.

Requirement	Verification procedure	Result
Hold the left and right page stacks flat during a page-turn cycle.	Engage the paperweights on books across the tested range and observe whether the pages stay flat.	Verified qualitatively; pages held flat during tested operation, but holding force was not measured.
Accommodate page sizes from ISO A5 to ISO A4.	Place A5 and A4 books on the workstation and confirm the mechanism reaches and turns the pages.	Verified qualitatively for tested A5-to-A4 placements; initial book placement is manual rather than automatically centered.

Table A.3 Power and electronics.

Requirement	Verification procedure	Result
Provide a stable logic supply to the Raspberry Pi during operation.	Run the full system and observe the Raspberry Pi for supply-related faults or resets.	Verified qualitatively; supply stable in operation.
Supply the 12 V motor rail without anomaly under the full actuator load.	Operate the steppers, gear motor, and servos together and observe the 12 V rail.	Verified qualitatively; no anomaly observed under combined actuator operation. Ripple, voltage sag, and peak current were not measured.

Table A.4 Vision and OCR pipeline.

Requirement	Verification procedure	Result
Capture the open page and extract its text with at least 90 % character accuracy.	Image a controlled ground-truth page, run the OCR service, and compare the returned text against the ground truth.	Pipeline verified end to end on the development machine; controlled ground-truth character accuracy and capture-to-text latency pending.
Return the recognized text for speech output.	Confirm the OCR service returns text to the Raspberry Pi for the speech stage.	Verified on the development machine.

Table A.5 Voice command and audio output.

Requirement	Verification procedure	Result
Recognize the four spoken commands with at least 80 % accuracy.	Speak each command in repeated trials and count correct recognitions.	Commands recognized correctly in informal testing; formal count pending.
Initiate the corresponding action within 3 s of the end of speech.	Measure the time from end of speech to the start of the commanded action.	Pending; measure from the end of the spoken command to the first visible mechanical action or software response.
Play the synthesized speech back to the user.	Issue a read command and confirm audible playback of the page text.	Speech synthesis verified on the development machine; powered speaker pending installation.

Appendix B Additional Photographs

Figures B.1 through B.7 present additional computer-aided design renderings and photographs of the prototype.

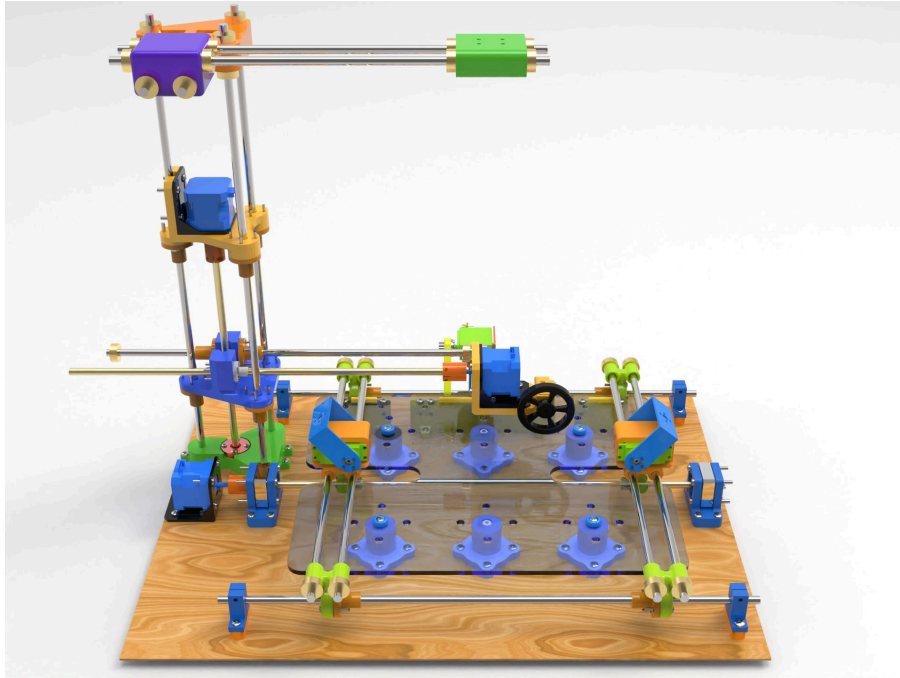


Figure B.1 Alternative view of the computer-aided design model.

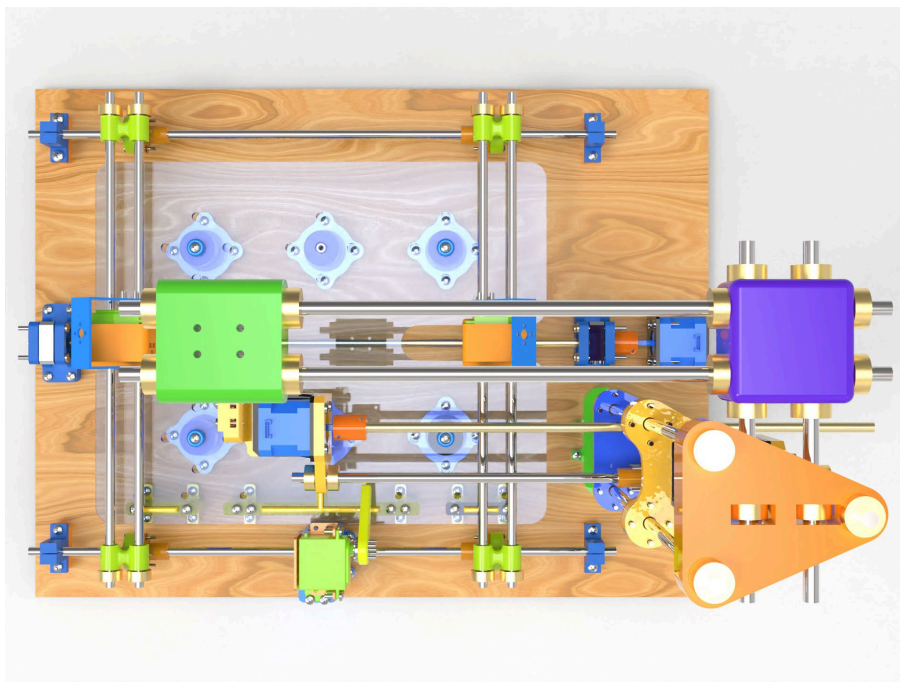


Figure B.2 Top view of the workstation showing the motion stages and paperweight positions.

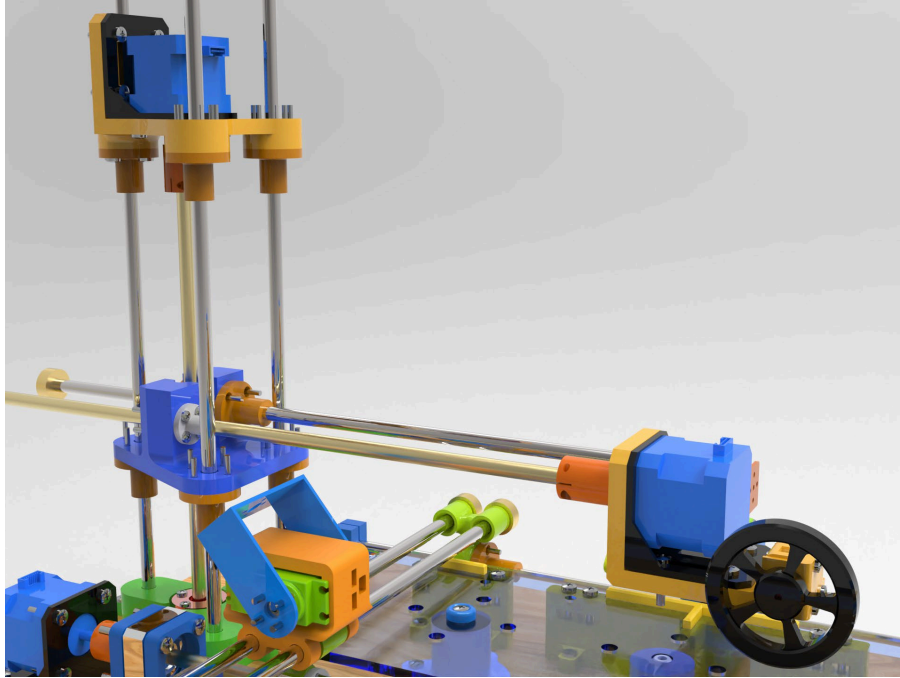


Figure B.3 Friction-wheel horizontal motion stage.



Figure B.4 Friction-wheel vertical motion axis.

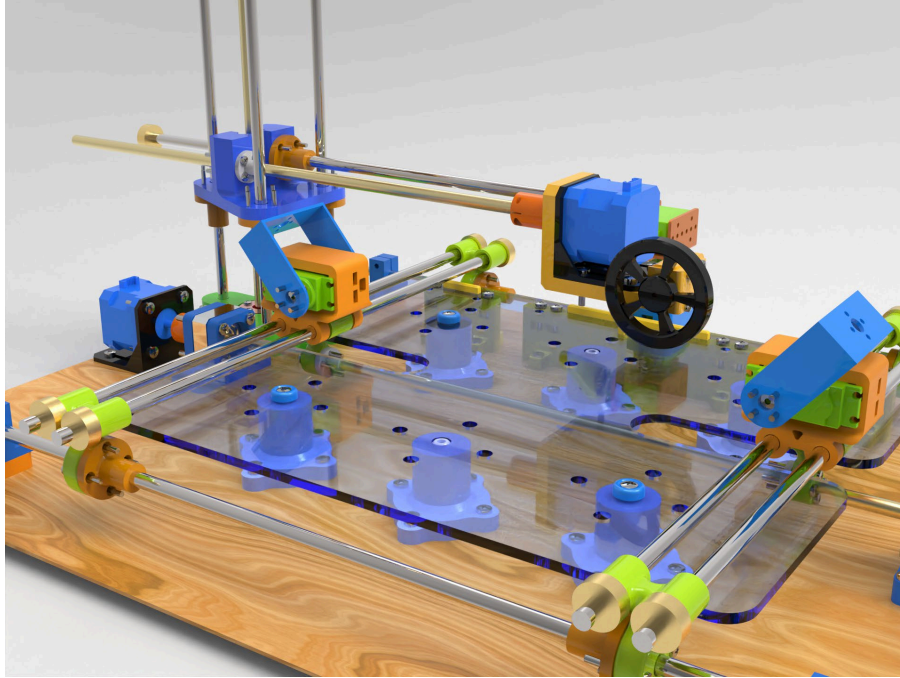


Figure B.5 Detail of the core functional area.

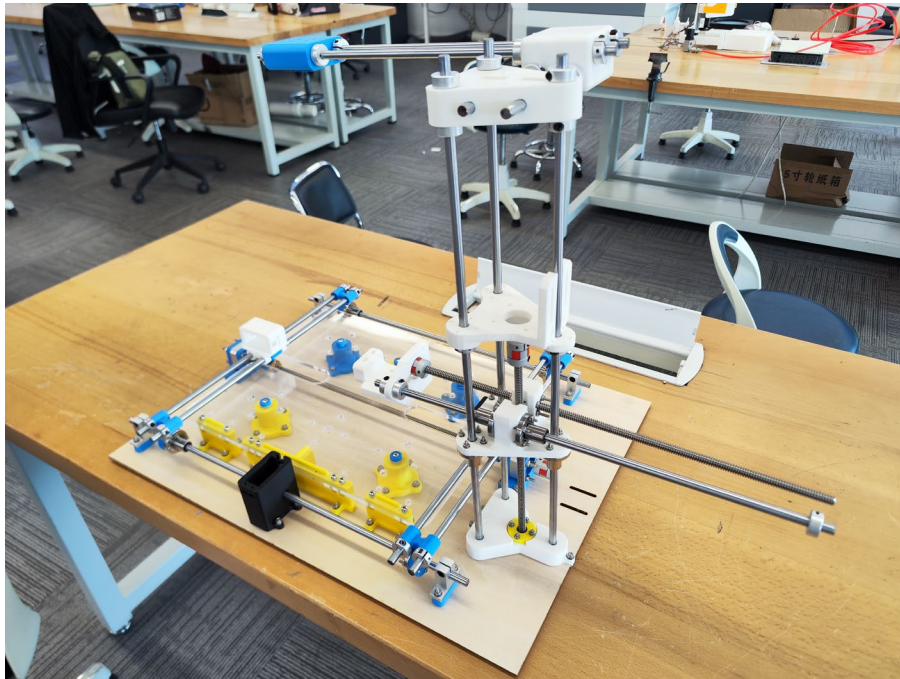


Figure B.6 Prototype during assembly.

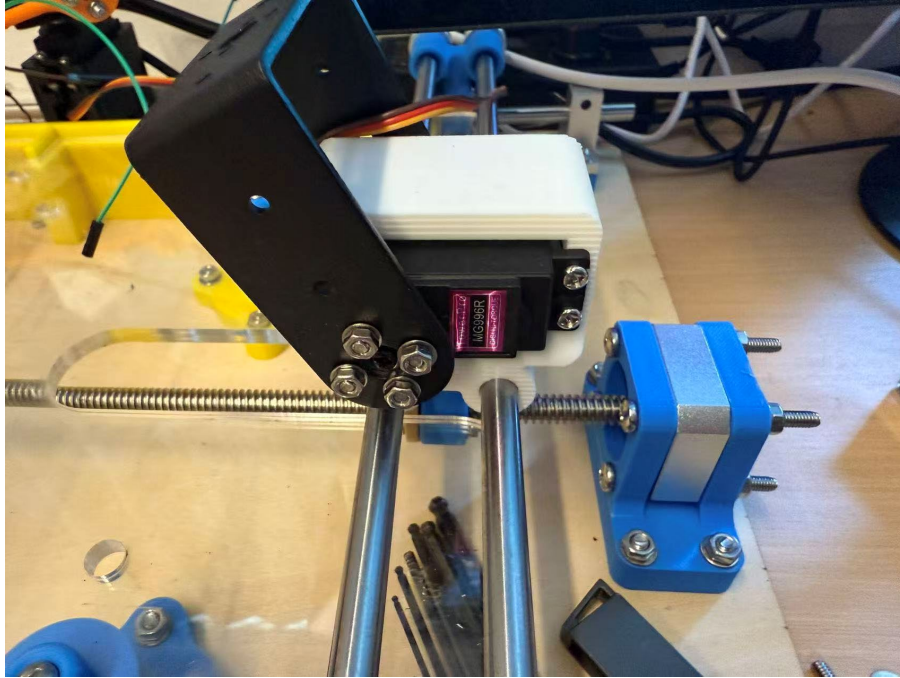


Figure B.7 Second paperweight arm.