

ROBOT DESK LAMP WITH USER INTERACTION

Design Document

ECE 445 Senior Design — Fall 2026

Team 23

Chris Shan (chriss4)

Yiming Niu (yimingn4)

Qianyu Li (qianyu4)

Project No. 23

Professor: Cunjiang Yu

TA: Wesley Pang

Date: October 2026

Contents

1. Introduction.....	3
1.1 Problem.....	3
1.2 Solution.....	3
1.3 Visual Aid.....	4
1.4 High-Level Requirements.....	4
2 Design.....	5
2.1 Block Diagram.....	5
2.2 Physical Design.....	5
2.3 Vision and User-Interaction Subsystem.....	6
2.4 Embedded Control and Communication Subsystem.....	7
2.5 Mechanical Structure and Servo Actuation Subsystem.....	8
2.6 Power Distribution and Custom PCB Subsystem.....	10
2.7 Lighting Subsystem.....	11
2.8 Software / Control Flow.....	12
2.9 Tolerance Analysis.....	13
3. Cost.....	15
3.1 Parts.....	17
3.2 Labor.....	18
3.3 Schedule.....	18
4. Conclusion.....	19
5. References.....	22

1. Introduction

1.1 Problem

Conventional desk lamps require the user to physically reposition the lamp whenever the illuminated work area changes. This is inconvenient during activities in which one hand is occupied, such as writing, soldering, assembly, drawing, or handling tools. Fixed lamps solve this problem only for a narrow region of the desk, while articulated lamps still require manual manipulation of the arm and lamp head. The project therefore addresses a human-machine interaction problem: how to move a light source to a user-selected target on a desk with minimal physical effort while preserving safe, stable illumination.

The proposed system is a robotic desk lamp whose articulated arm automatically repositions the light. The user indicates a desired location using a single-hand interaction observed by a camera. Rather than requiring the user to manually bend several joints, the system converts the visual target into a lamp-head target pose, computes the required joint angles, and commands the arm to move. This approach combines perception, embedded control, power electronics, and mechanical actuation in one integrated product.

1.2 Solution

The current design uses a four-degree-of-freedom articulated lamp driven by four 12 V ST3215 serial-bus servos. A Raspberry Pi and camera perform the higher-level perception task. The Raspberry Pi converts the camera image into a target position or direction and transmits that target to an ESP32-S3 over a 3.3 V UART link. The ESP32-S3 executes low-level control, including inverse-kinematics calculations, joint-limit enforcement, servo communication over the shared half-duplex bus, and LED pulse-width modulation (PWM).

A custom PCB distributes 12 V power from an external high-current supply, provides independent fused branches for the four servos, LED, Raspberry Pi, and ESP32 logic rail, converts 12 V to 5 V for the Raspberry Pi using a TPS566231 synchronous buck regulator, converts 12 V to 3.3 V for the ESP32 using a TPS54202H synchronous buck regulator, and switches the 12 V lamp with an AO3400A N-channel MOSFET. Mains voltage is intentionally kept off the custom PCB; the board receives only low-voltage DC from an external AC/DC supply.

1.3 Visual Aid

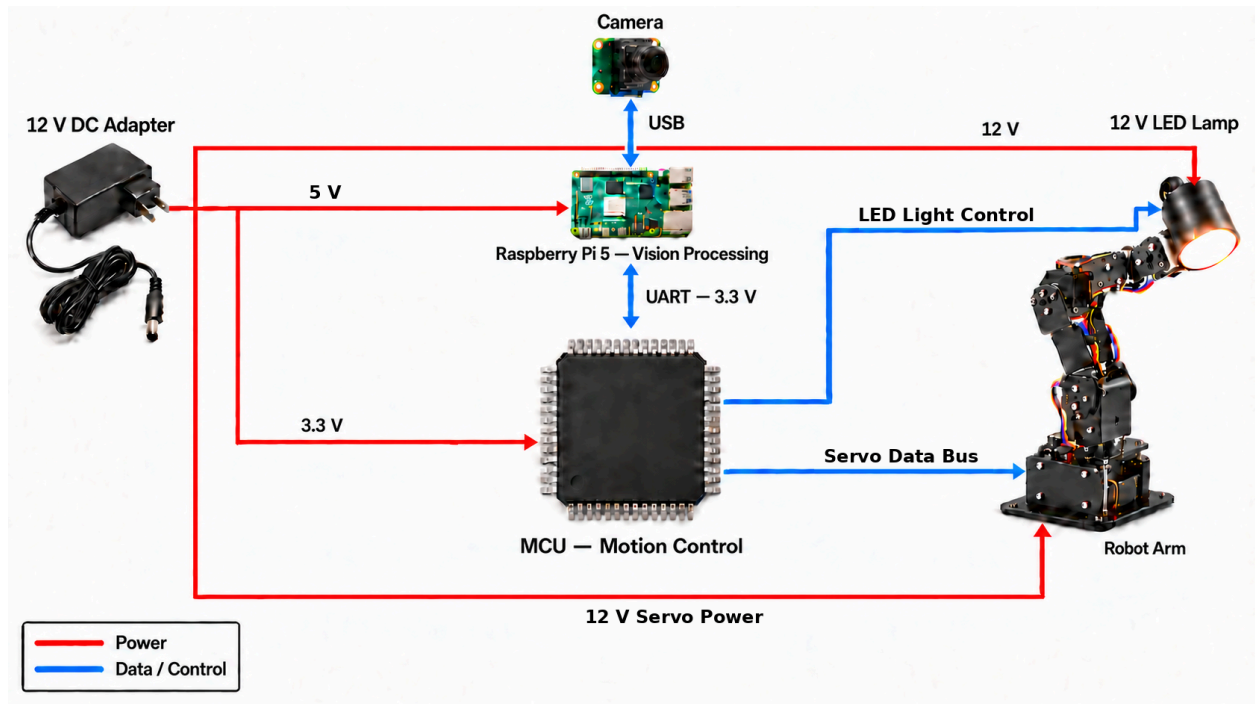


Figure 1 - Visual Aid of Design

Figure 1 provides a pictorial overview of the proposed hardware architecture. It shows the 12 V DC adapter supplying the Raspberry Pi 5, microcontroller, servo-powered robot arm, and LED lamp; the camera connected to the Raspberry Pi for vision processing; UART communication between the Raspberry Pi and microcontroller; and PWM control from the microcontroller to the robot arm and lamp.

1.4 High-Level Requirements

The project will be considered successful if all of the following system-level requirements are satisfied:

1. The robotic lamp shall successfully reposition its lamp head toward a user-specified target direction in at least 80% of test trials conducted within the defined desk workspace, with a response time of 15 seconds or less.
2. The vision and interaction subsystem shall correctly interpret the user's intended gesture or target direction and complete the corresponding action in at least 80% of test trials under the defined operating conditions.
3. The complete system shall operate the Raspberry Pi, ESP32-S3 microcontroller, LED lamp, and all four servo motors simultaneously for at least 6 minutes without processor reset, microcontroller reset, or loss of servo control, while the 12 V, 5 V, and 3.3 V supply rails remain within $\pm 5\%$ of their nominal values.

2 Design

2.1 Block Diagram

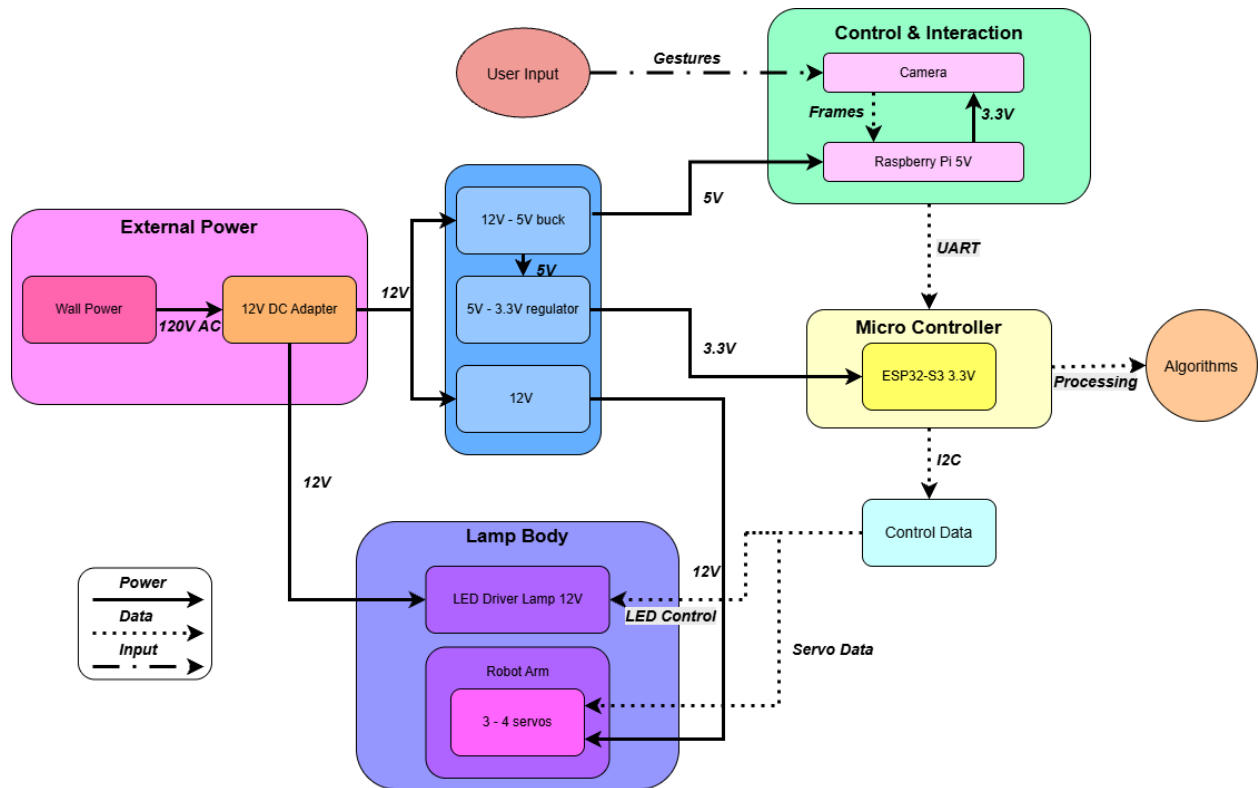


Figure 2 - Block diagram

The system is partitioned so that computationally heavy perception runs on the Raspberry Pi while deterministic low-level motion and power-control tasks remain on the ESP32-S3. This separation prevents computer-vision processing load from directly affecting servo timing and gives the system a simple fallback behavior if the perception stream is lost. The custom PCB forms the electrical integration point for power conversion, overcurrent protection, servo connectivity, logic control, and LED dimming.

2.2 Physical Design

The mechanical assembly is a four-joint articulated desk-lamp arm. The joints provide base rotation and additional articulated segments that position and orient the lamp head over the desk. Four ST3215 servos are used because the servo family supports serial addressing on a shared single-wire TTL bus, provides position feedback, and is rated for operation from approximately 9 V to 12.6 V. Waveshare lists 30 kg·cm stall torque at 12 V and 2.7 A stall current for the ST3215 used in its servo assemblies.

The camera must be mounted so that its field of view covers the entire interaction workspace without being occluded by the arm. The custom PCB and Raspberry Pi should be located near the base so that heavy electronics do not increase the moving-link inertia. The external 12 V supply must be enclosed so that mains terminals are not user-accessible.

The lamp is built on an 18 cm × 18 cm × 5 cm base that houses the custom PCB, the Raspberry Pi 5, and a 160 mm × 160 mm × 2.5 mm steel ballast plate, bringing the base mass to approximately 1 kg. The lower arm measures 25 cm and the upper arm measures 18 cm between joint centers, and the lamp head extends approximately 10 cm from the head-tilt joint to the front of the shade. In its upright pose the lamp stands approximately 51 cm tall, and at full horizontal extension it reaches approximately 53 cm from the base-rotation axis. Each of the four STS3215 servos measures 45.2 mm × 24.7 mm × 35 mm and weighs approximately 55 g, and the estimated worst-case static torque on the shoulder joint is 9.4 kg·cm, giving a safety factor of about 3.2 against the 30 kg·cm stall torque. The ballasted base exceeds the approximately 760 g required to resist tipping at full extension, and joint limits in firmware further prevent the arm from reaching the fully horizontal position during normal operation.



Figure 3 - 3D model of Robot Lamp with key components including LED, camera, and motors

2.3 Vision and User-Interaction Subsystem

The vision and user-interaction subsystem is responsible for converting the user's single-hand interaction into a target that can be used by the lamp controller. The subsystem consists of a camera connected to a Raspberry Pi. The camera continuously observes the defined desk workspace, while the Raspberry Pi performs the image-processing and interaction-recognition tasks.

The desired lighting target is constrained to the known tabletop plane. Therefore, the system does not attempt to reconstruct an arbitrary three-dimensional hand position from a single monocular camera image. Instead, the perception subsystem detects the user's hand within the image and extracts a selected image-space feature representing the desired target location. This feature is then mapped to a two-dimensional position on the tabletop, x_{target} and y_{target} .

The exact hand feature used to define the target, such as a fingertip or hand centroid, will be selected during implementation and calibration.

In parallel with target-position detection, the Raspberry Pi will classify a small set of predefined hand gestures. Target position and gesture recognition serve different functions: the target position specifies where the lamp should point, while the gesture specifies what action the user is requesting, such as enabling tracking or commanding the lamp to hold its current position. The final gesture set and classification method have not yet been selected.

Figure 4 shows the planned perception pipeline. A camera frame is acquired by the Raspberry Pi and processed to identify the user's hand. The resulting hand information is used by two parallel processing paths: one determines the target location within the desk workspace, while the other determines the interaction command. The Raspberry Pi validates the detected target and gesture before transmitting the resulting command to the ESP32-S3 over a 3.3 V UART interface.

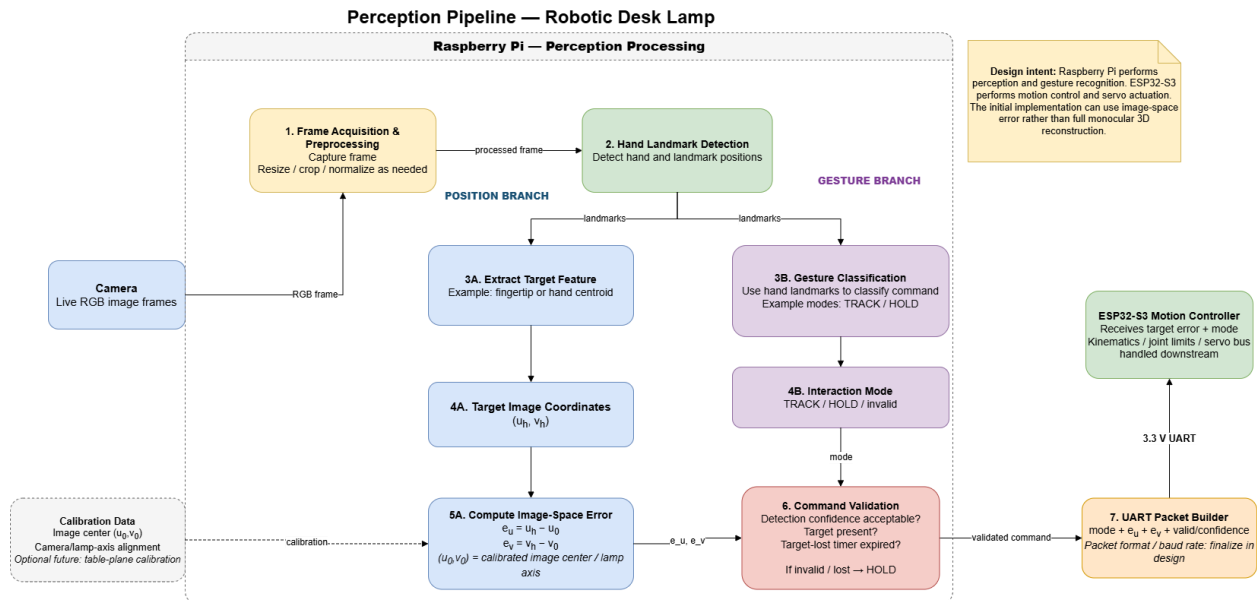


Figure 4. Perception pipeline for extracting a tabletop target location and interaction command from the camera image and transmitting the resulting information to the ESP32-S3.

This division of computation is intentional. Image processing and gesture recognition are assigned to the Raspberry Pi because these operations require substantially more computational capability than the low-level motion-control tasks. The ESP32-S3 remains responsible for the kinematic calculations, joint-limit enforcement, and communication with the four servo motors. The interface between the two processors therefore separates perception from actuation.

For the baseline implementation, the perception subsystem will use a Raspberry Pi 5 with 1 GB of RAM and a Raspberry Pi Camera Module 3 Wide. This combination is selected to provide enough processing capability for real-time hand detection and gesture recognition while maintaining a relatively low system

cost. The wide-angle camera is intended to increase coverage of the desk workspace and reduce the likelihood that the user's hand leaves the field of view during interaction.

The Raspberry Pi will transmit the processed target information and gesture command to the ESP32-S3 over UART. The exact packet format will be finalized during implementation, but it will contain the target information required by the motion controller together with the current interaction command.

2.4 Embedded Control and Communication Subsystem

The ESP32-S3-WROOM-1-N8 is the low-level controller on the custom PCB. Espressif specifies a recommended supply range of 3.0–3.6 V for the ESP32-S3-WROOM-1 family [1]. The PCB therefore provides a dedicated 3.3 V rail. The ESP32 receives target information from the Raspberry Pi through a direct 3.3 V UART connection: Pi TX connects to ESP RX, Pi RX connects to ESP TX, and both systems share GND. Because both sides use 3.3 V logic, no level translation is required for the UART interface.

The ESP32 transforms the target into joint commands using inverse kinematics and applies joint limits before transmitting commands over the shared half-duplex serial servo bus. The servo connectors use pin 1 = DATA, pin 2 = +12 V, and pin 3 = GND. The lamp brightness is independently controlled by an ESP32 PWM output driving the LED MOSFET gate through a 100 Ω series resistor, with a 100 k Ω pulldown ensuring that the MOSFET remains off while the GPIO is high impedance.

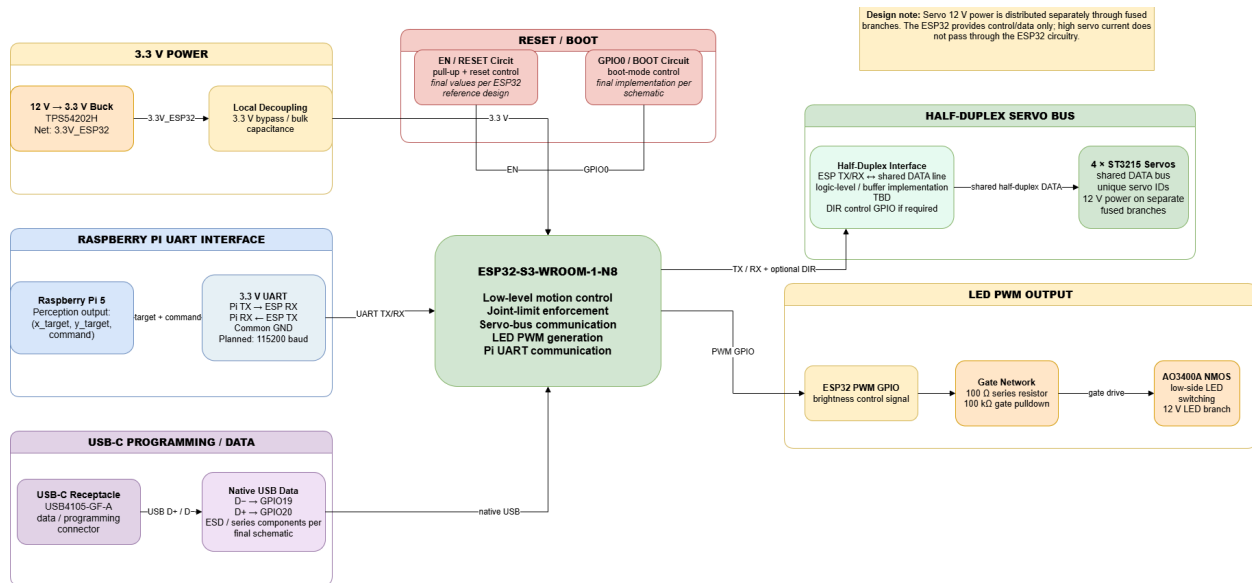


Figure 5. ESP32 control and communication schematic

Requirement	Verification
The Pi→ESP32 communication link shall transfer target packets for 10 continuous minutes with a packet error rate below 1% and without ESP32 reset.	Equipment: Raspberry Pi, ESP32 PCB, serial logging. Test: transmit a known counter/CRC packet stream at the final baud rate for 10 minutes. Count missing or corrupted packets. Demonstration: log and computed packet error rate.
The ESP32 shall command all four servos independently on the shared half-duplex bus and shall prevent commands outside the configured mechanical joint limits.	Test: command each joint to minimum, midpoint, and maximum allowed positions while monitoring the other joints. Attempt software commands outside each limit. Demonstration: joint-angle table showing commanded and accepted/limited positions.

Requirement	Verification
The LED PWM output shall default to OFF during ESP32 reset or high-impedance startup.	Equipment: oscilloscope/DMM. Test: power-cycle and reset the ESP32 while monitoring MOSFET gate and lamp current. Demonstration: waveform showing gate held low by 100 k Ω pulldown.

2.5 Mechanical Structure and Servo Actuation Subsystem

Four ST3215 smart servos provide the mechanical actuation of the robotic lamp. The servos share a common DATA bus and ground reference, while each servo receives 12 V power through its own 3 A slow-blow fuse. This provides branch-level fault isolation without requiring a separate detailed electrical figure in the mechanical section. The selected board connector is the Molex 22035035 three-position header, which matches the servo connector family and pin order used in the design.

At 12 V, the specified stall current is approximately 2.7 A per servo, giving a worst-case simultaneous stall current of approximately 10.8 A for all four servos [2]. During normal operation, the ESP32-S3 will enforce calibrated joint limits and motion profiles to reduce the likelihood of prolonged mechanical stall or operation beyond the intended workspace.

Figure 7 shows the side view of the robotic lamp and identifies the four servo-driven joints used for positioning the lamp head. A complete 3D CAD model of the mechanical structure has already been developed. Detailed mechanical dimensions are omitted from this document for clarity; the CAD model can be provided if additional dimensional or assembly information is required.

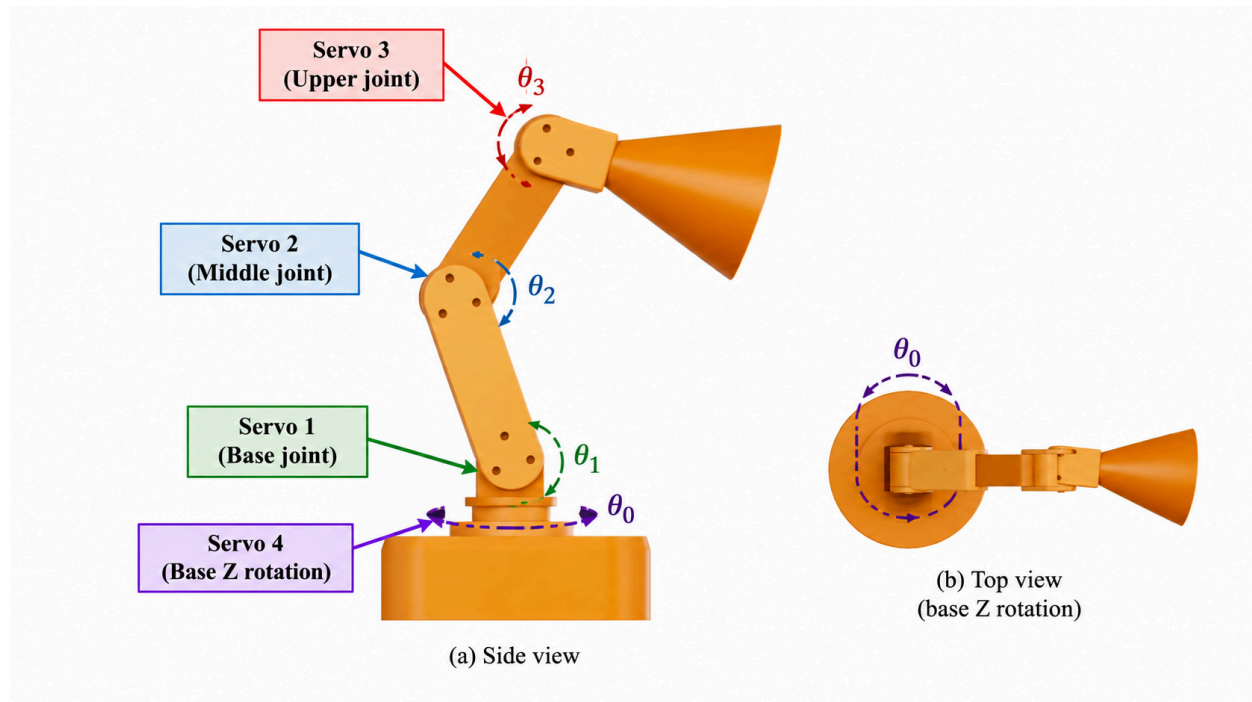


Figure 6. Mechanical joint layout of the robotic lamp, showing the three planar arm joints and the base Z-axis rotation driven by the four ST3215 servos.

Requirement	Verification
All four joints shall move together through the defined trajectory without mechanical collision, cable binding, or sustained stall.	Test: execute a trajectory that traverses the boundary of the defined workspace at least five times. Observe current, cable motion, and mechanical clearance. Demonstration: video plus test checklist.
For at least 8 of 10 target trials, the servo system shall settle the lamp head at the requested target pose within the final position tolerance.	Equipment: marked desk grid or camera-based measurement. Test: command 10 targets across the workspace and measure final light-spot position. Demonstration: requested vs. measured target table.
Each servo branch fuse shall remain intact during normal motion but shall isolate a branch fault without opening unrelated servo-branch fuses.	Equipment: current-limited bench supply or controlled fault fixture. Test: verify normal peak current stays below the 3 A branch rating, then simulate a branch overcurrent under supervised conditions. Demonstration: branch-current measurements and fault-isolation result.

2.6 Power Distribution and Custom PCB Subsystem

The power subsystem supports the simultaneous operation requirement in Section 1.4 by distributing low-voltage DC to four 12 V ST3215 servos, the Raspberry Pi and camera, the ESP32-S3, and the lamp. We plan to use one enclosed external 12 V AC/DC adapter. The custom PCB will receive the adapter output through an input connector and distribute it through protected branches. The servo and compatible 12 V lamp branches use the input voltage; separate buck converters provide the nominal 5 V Pi rail and 3.3 V logic rail. Converter selection, protection values, and connector ratings must be checked against the current EasyEDA schematic before fabrication. Part numbers in the existing introduction remain candidate selections until that check is complete.

The Pi branch will terminate in a dedicated power connector and short cable to the Pi's 5 V and ground header contacts if the retained regulator passes load testing. This is the proposed direct-header supply route discussed for the prototype. It requires correct polarity, suitable contacts and wire, and protection against accidental contact with adjacent header pins. It must never connect the 12 V branch to the Pi header. A USB-C power source with PD capability would be a separate design option; adding PD signaling to the header supply would not improve its voltage or current capacity. The final connector, header contacts, current rating, and power arrangement remain open. We will use one intentional power source for the Pi during testing and review any simultaneous external connections for backfeeding [3].

We will size the adapter from simultaneous branch loads rather than from the sum of regulator nameplate ratings. A preliminary input-current model is:

$$I_{\text{adapter}} = \sum_k I_{\text{servo},k} + \frac{P_{\text{lamp}}}{12} + \frac{V_{\text{Pi}} I_{\text{Pi}}}{12 \eta_{\text{Pi}}} + \frac{3.3 I_{\text{logic}}}{12 \eta_{\text{logic}}}$$

Here η_{Pi} is measured or specified conversion efficiency, and the sum includes all four servos. The selected 12 V ST3215 is listed with a 2.7 A locked-rotor current [2], so four locked servos alone represent approximately 10.8 A. This is a fault/loading case, not an assumed continuous operating current. For illustration, a 5 V, 5 A Pi load at an assumed 90% conversion efficiency adds 2.31 A at 12 V. Thus that combined case exceeds 13.11 A before including the lamp and ESP32. Lamp power, motion current, startup peaks, adapter transient behavior, and the intended overload response must be established

before selecting an adapter or fuse. Fuse selection must coordinate its time-current curve with the weakest wire, contact, and PCB path it protects.

PCB layout will keep regulator switching loops short, place decoupling near the relevant pins, and separate high-current servo returns from sensitive feedback and communication paths while maintaining a common ground reference. We will route the high-current branches using copper width, thickness, and vias selected for the measured load and acceptable temperature rise. Logic connectors will carry 3.3 V UART signals and ground; the serial-servo bus will use the required half-duplex interface. USB used to program the ESP32 will be treated as a data connection while the board is powered by its normal input, unless the current schematic explicitly supports USB-only power. Footprints, protection polarity, and connector pinouts will be reviewed before schematic-to-PCB synchronization and design-rule checks.

Requirement	Verification
During at least 6 minutes of combined operation, the nominal 12 V, 5 V, and 3.3 V rails remain within the Section 1.4 limits of 11.4–12.6 V, 4.75–5.25 V, and 3.135–3.465 V at the loads, with no controller reset or loss of servo control.	Measure at the servo, Pi, and ESP32 supply contacts with an oscilloscope during concurrent motion, camera processing, and maximum selected lamp output. Record extrema and reset/error logs. The startup interval is evaluated separately.
Each branch is correctly polarized and has protection compatible with its current, wiring, and contacts.	Check the current netlist and connector pinout, then power branches separately using a current-limited supply. Use a controlled dummy load to check the selected protection response; do not short an assembled board by hand.
Power connections remain within their specified electrical and thermal ratings under the selected continuous and transient load cases.	Measure current and temperatures at regulators, input connector, branch connectors, and wiring. Compare with the actual part and material ratings; establish the maximum ambient temperature and test duration before acceptance.

2.7 Lighting Subsystem

The lighting subsystem provides useful illumination while the arm directs the lamp head toward the selected desk area. The current concept uses an ESP32 PWM output to control a transistor switch in the lamp's return path. With a compatible two-wire 12 V lamp, current flows from the protected 12 V lamp branch through the lamp and the N-channel MOSFET to ground. A series gate resistor controls the gate charging current, and a gate pulldown establishes an off state while the MCU pin is unconfigured. The existing document identifies AO3400A as a candidate switch; its suitability depends on the final lamp current, gate drive, copper area, and temperature [4].

This switching circuit controls power delivery but does not regulate LED current. A 12 V lamp module containing current limiting may support supply PWM if its manufacturer permits it. A bare high-power LED instead requires a suitable constant-current driver, with a PWM/DIM interface or another approved dimming method. A module with internal input capacitors or a shared signal ground may also be unsuitable for low-side supply switching. We must select the lamp before deciding whether to retain this topology or add a driver. Lamp voltage, power, current, beam pattern, heatsink, and dimming compatibility remain unresolved.

For a compatible load, the ESP32 LEDC peripheral will produce the PWM waveform [1]. We propose 5 kHz and 10-bit duty resolution as initial bench settings, subject to driver compatibility, gate-switching

measurements, audible noise, and camera banding. The duty increment is approximately $1/1024$, or 0.098%; this is electrical control resolution and does not establish equivalent optical brightness accuracy. Firmware will support an explicit off command and a bounded brightness setting. The GPIO must match the actual schematic net; GPIO4 is a previously discussed candidate, not a verified assignment. The startup command will be off until initialization is complete. Whether a separate physical lamp switch is needed remains a user-interface decision.

At full output, a first conduction-loss estimate is $P_{\text{cond}} = I_{\text{lamp}}^2 R_{\text{DS(on)}}$. For a hypothetical 12 W lamp at 12 V, $I_{\text{lamp}} = 1$ A. Using the AO3400A's 48 milliohm maximum specified at $V_{\text{GS}} = 2.5$ V and 25 degrees C gives approximately 0.048 W before switching losses and temperature effects [4]. This illustrative load is not the selected lamp. We will measure drain and gate waveforms, verify that the switch turns fully on and off, and measure the assembled lamp and transistor temperatures. A low calculated conduction loss alone does not establish thermal suitability.

Requirement	Verification
The lighting output responds to an off command and a monotonic sequence of brightness commands.	Measure PWM and relative illumination at a fixed desk position for off and several ordered duty settings. Record any nonlinear response or flicker.
At the agreed working distance, the illuminated area and illuminance meet the selected desk-task requirement.	Measure lux and beam coverage on a defined grid with the lamp at representative arm poses. Required lux, grid size, working distance, and coverage remain to be selected.
Lamp, driver, switch, wiring, and accessible surfaces remain below their agreed component and user-contact temperature limits.	Run the maximum permitted lamp output with simultaneous system loading, measure temperatures, and compare with supplier limits and the assembly's safety targets.

2.8 Software / Control Flow

We will divide software between the Raspberry Pi perception program and the ESP32 controller. The Pi handles camera capture, hand/gesture interpretation, and transformation of the observed interaction into a calibrated target. The ESP32 validates the target, solves the arm kinematics, enforces joint and motion limits, communicates with the four serial-bus servos, and controls lamp PWM. ST3215 motion is commanded through addressed serial packets on the shared half-duplex bus; it is not commanded by the lamp's PWM signal. This arrangement supports the interpretation and repositioning requirements in Section 1.4 while allowing the ESP32 to reject unsafe requests independently of vision processing.

The proposed control sequence is initialization, idle, target acquisition, validation, motion, and settled operation. Initialization sets lamp output off, loads calibration and motion limits, checks communication with each unique servo ID, and waits for a deliberate start. Automatic homing cannot be assumed safe until a clear path and joint references are established. During target acquisition, the Pi will reject detections outside the calibrated workspace and require a stable observation before issuing a target. Camera placement, target representation, gesture definitions, confidence threshold, and the stability interval must be fixed with the vision implementation.

The proposed Pi-to-ESP32 link is a crossed TX/RX 3.3 V UART connection with a common ground. Initial settings are 115200 baud, eight data bits, no parity, and one stop bit. A compact packet will contain a start marker, protocol version, message type, sequence number, payload length, target or brightness data, and a checksum/CRC. The final byte layout and CRC choice remain to be agreed by both firmware owners. A proposed 32-byte command takes $32 \times 10 / 115200 = 2.78$ ms to transmit with 8N1 framing, which is a small portion of the 15-second system response limit. This calculation excludes perception time, queueing, bus feedback, and mechanical travel. The Pi UART rate is independent of the ST3215 bus rate, which must match the actual servo configuration.

The ESP32 parser will check framing, declared length, checksum, sequence number, and payload bounds before accepting a request. A coordinate target must use agreed units and the calibrated arm frame. The kinematics solver will reject unreachable poses and solutions outside the joint limits. Commands will be converted into a speed- and acceleration-limited trajectory rather than abrupt unrestricted joint changes. Feedback from addressed servos will be used to distinguish command acceptance from motion completion. A newer command will be accepted according to an agreed cancellation/update policy so that the system does not accumulate a queue of obsolete visual targets.

We propose a heartbeat every 100 ms and a communication timeout of 500 ms as initial integration settings. These are software proposals, not established safety limits. On a timeout, malformed command stream, servo communication fault, or failed initialization, the controller will inhibit new motion and enter a fault state. The final fault behavior must be checked on the assembled arm: retaining holding torque may support the arm, while removing torque may allow it to drop. Lamp behavior on a fault and the method of operator recovery must also be selected. Recovery should require a new valid start sequence; old movement commands should not resume automatically after a reboot or reconnection.

We will timestamp image acquisition, target transmission, command acceptance, motion start, and settled completion. A provisional allocation is 2 seconds for interpretation and stable-target selection, 0.5 seconds for communication and validation, and 12.5 seconds for motion and settling. These allocations sum to the existing 15-second limit and will be revised using measured results. We must define the angular or desk-position tolerance for a successful action, the settling criterion, the workspace, lighting conditions, and the number of trials before the existing 80% success requirement can be verified reproducibly.

Requirement	Verification
At least 80% of the defined interaction and repositioning trials succeed within 15 seconds, consistent with Section 1.4.	Test a documented set of targets and gestures across the defined workspace and lighting conditions. Log every trial, timestamps, target error, and failure reason. Fix the trial count and success tolerance before testing.
Corrupt, repeated, stale, out-of-range, and unreachable commands do not initiate a new motion.	Inject each case over the Pi UART and verify rejection using acknowledgements, servo telemetry, and observation. Test checksum errors and truncated packets.

Requirement	Verification
Loss of the Pi stream or servo communication causes the agreed fault response and does not automatically restart motion on recovery.	Disconnect communication during controlled, supported-arm tests; verify timeout, fault status, torque behavior, lamp behavior, and explicit recovery.

2.9 Tolerance Analysis

We select the Pi supply path as the main tolerance analysis because a voltage disturbance can interrupt perception and prevent the lamp from completing a requested action. The relevant requirement is the nominal 5 V rail staying within 4.75–5.25 V during combined operation. We analyze regulator set-point variation, cable and contact resistance, and a load transient. This follows the course approach of identifying a critical function, propagating component tolerances, and using the result to constrain the design [5]. The TA should confirm that this is the appropriate primary analysis.

The following is a worked design case, not a claim about the current EasyEDA component values. It assumes a TPS566231, a 220 kilohm upper feedback resistor R_T , a 30 kilohm lower resistor R_B , and a 5 A Pi branch load. The actual regulator, divider values and tolerances, and Pi load must be verified. For the assumed regulator, the reference range is 0.591–0.609 V over its specified junction-temperature range [6]. Ignoring additional regulation errors initially:

$$V_{\text{out}} = V_{\text{FB}} \left(1 + \frac{R_T}{R_B} \right)$$

$$V_{\text{out,min}} = V_{\text{FB,min}} \left[1 + \frac{R_T(1 - t_R)}{R_B(1 + t_R)} \right]$$

$$V_{\text{out,max}} = V_{\text{FB,max}} \left[1 + \frac{R_T(1 + t_R)}{R_B(1 - t_R)} \right]$$

The nominal output is $0.600 (1 + 220/30) = 5.000$ V. Following table compares independent worst-case resistor limits; t_R is the tolerance of each resistor. These calculations use the full-temperature reference range rather than its narrower 25-degree value.

Resistor tolerance assumed	Minimum set point	Maximum set point	Margin above 4.75 V
1%	4.839 V	5.165 V	89 mV
0.1%	4.916 V	5.084 V	166 mV

The Pi sees less voltage than the regulator output when current flows through the positive and return conductors, contacts, and PCB path. Let R_{loop} include that complete loop resistance and let ΔV_{neg} include the worst negative ripple, transient response, and regulation error not already included in the reference bound. Then:

$$V_{Pi,min} = V_{out,min} - I_{Pi}R_{loop} - \Delta V_{neg}$$

$$R_{loop,max} = \frac{V_{out,min} - 4.75 - \Delta V_{neg}}{I_{Pi}}$$

For an assumed 50 mV additional negative-error allowance and a 5 A load, the 1% resistor case requires R_{loop} no greater than 7.84 milliohms. A hypothetical 20 milliohm path drops 0.100 V and gives $V_{Pi,min} = 4.839 - 0.100 - 0.050 = 4.689$ V, which fails the project limit. This shows why a 6 A regulator rating alone cannot establish a suitable Pi supply.

With assumed 0.1% resistors, the same 20 milliohm path and 50 mV allowance instead give $4.916 - 0.100 - 0.050 = 4.766$ V. For a 50 mV positive overshoot allowance at light load, the upper bound is $5.084 + 0.050 = 5.134$ V. This case meets the project voltage window conditionally, but has only about 16 mV of remaining low-side margin. It establishes a feasible combination of assumptions; it does not select new resistors or demonstrate that the present PCB meets those assumptions. Lower resistance, tighter set-point accuracy, or a revised nominal voltage may be needed after measurements. Any revised nominal voltage must also satisfy the upper-voltage limit.

Output-capacitor behavior supplies a further constraint. During a brief interval before the regulator

supplies the extra load, an approximate capacitor droop is $\Delta V_C = \frac{\Delta I \Delta t}{C_{effective}} + \Delta I \cdot ESR$. For an illustrative 1 A step lasting 10 microseconds and 110 microfarads of effective capacitance, the capacitive term alone is 90.9 mV. Holding that term to 50 mV would require at least 200 microfarads of effective capacitance before allowing for ESR. Nominal ceramic capacitance may be reduced by DC bias, tolerance, and temperature. These illustrative values must be replaced by the measured load step, response interval, effective capacitor data, and the regulator's permitted output-network range before changing the circuit.

We will measure the regulator output and the Pi supply contacts simultaneously under camera processing and four-servo motion. A controlled electronic-load step will test the Pi branch before full assembly. The test will establish loop resistance, minimum and maximum voltage, ripple, recovery, temperatures, and any undervoltage/reset indication. Acceptance requires the current divider and output network to meet the voltage window over the selected operating conditions. If they fail, the measured dominant error will determine the required circuit or wiring change.

Power integrity does not by itself prove aiming accuracy. Once mechanical dimensions are available, we will also propagate camera-calibration error, joint zero error, backlash, link deflection, and joint

quantization through the arm kinematics. The supplier's 4096-position encoding implies a nominal angular step of $360/4096 = 0.0879$ degrees, with half-step quantization of 0.0439 degrees [2]. At an assumed 0.4 m lever arm, one joint's half-step contributes about 0.31 mm. Actual positioning error can be much larger; encoder resolution is not a specification of loaded arm accuracy. The overall error should be evaluated using the pose Jacobian and measured joint errors against a defined lamp-position or beam-centroid tolerance.

3. Costs

3.1 Parts

Table 1 Parts Costs

Part	Manufacturer	Quantity	Unit Cost (\$)	Total Cost (\$)
Servo Motor	FEETECH (ST3215)	4	27.99	111.96
ESP32-S3	Espressif Systems	1	6.76	6.76
Raspberry Pi5	Raspberry Pi	1	100	100
Camera Module	Raspberry Pi	1	30	30
22 μ F Capacitor (1210)	Samsung	7	0.65	4.55
22 μ F Capacitor (0603)	Samsung	1	0.23	0.23
10 μ F Capacitor	Samsung	5	0.28	1.40
1 μ F Capacitor	Samsung	2	0.10	0.20
0.1 μ F Capacitor	Samsung	7	0.10	0.70
22nF Capacitor	Samsung	1	0.10	0.10
56pF Capacitor	Samsung	1	0.10	0.10
47pF Capacitor	Samsung	1	0.10	0.10
510k Ω Resistor	YAGEO	1	0.10	0.10
220k Ω Resistor	YAGEO	1	0.10	0.10
100k Ω Resistor	YAGEO	3	0.10	0.30
75k Ω Resistor	YAGEO	1	0.10	0.10

30k Ω Resistor	YAGEO	1	0.10	0.10
22.1k Ω Resistor	YAGEO	1	0.10	0.10
10k Ω Resistor	YAGEO	4	0.10	0.40
5.1k Ω Resistor	YAGEO	2	0.10	0.20
100 Ω Resistor	YAGEO	1	0.10	0.10
49.9 Ω Resistor	YAGEO	1	0.10	0.10
22 Ω Resistor	YAGEO	2	0.10	0.20
3-Pin Connector	Molex	5	0.22	1.10
2-Pin Power Connector	JST	2	0.15	0.30
2-Pin Terminal Block	DORABO	1	0.11	0.11
USB Type-C Connector	GCT	1	0.80	0.80
20A Through-Hole Fuse	Littelfuse	1	0.48	0.48
3A SMD Fuse	Littelfuse	6	1.96	11.76
1A SMD Fuse	Littelfuse	1	2.36	2.36
2.2 μ H Inductor	Vishay	1	1.13	1.13
10 μ H Inductor	Vishay	1	1.13	1.13
N-Channel MOSFET	AOS	1	0.52	0.52
Pushbutton Switch	TE Connectivity	2	0.13	0.26
Step-Down Converter	Texas Instruments	2 (TPS566231 + TPS54202H)	2.73 / 1.36	4.09
ESD Protection IC	Würth Elektronik	1	0.86	0.86
Single Buffer, 3-State	Texas Instruments	2 (LVC1G125 + LVC1G126)	0.94 / 0.92	1.86
Total				284.66

3.2 Labor

We estimate engineering labor at \$42 per hour for each team member. The initial estimate allocates six hours per week over fourteen working weeks to each member, including design, procurement, construction, software development, testing, and documentation. Table 2 summarizes the estimate using the course-required factor of 2.5 [5]. The team will revise the hours to reflect its actual workload before submission.

Labor cost per member = \$42/hour × 2.5 × 84 hours = \$8,820.

Table 2 Labor Costs

Team member	Rate per hour	Estimated hours	Multiplier	Estimated cost
Chris Shan	\$42	84	2.5	\$8,820
Yiming Niu	\$42	84	2.5	\$8,820
Qianyu Li	\$42	84	2.5	\$8,820
Total		252		\$26,460

This is an estimated engineering labor cost. Parts, fabrication, and any quoted machine-shop charges will be added after the updated design and bill of materials have been checked. The project total will be \$26,460 plus verified parts and fabrication costs; it cannot yet be stated as a complete dollar amount. Any shop labor charged separately must be counted once.

3.3 Schedule

Table 3 organizes the remaining design, construction, integration, and verification work around the published course milestones [5]. The dates describe planned work and do not establish that a task has been completed. Qianyu Li leads Raspberry Pi software, camera integration, and perception. Chris Shan leads the lamp body, mechanical design, and assembly. Yiming Niu (Tyler) leads ESP32 firmware and servo communication. Power and PCB integration, system testing, and documentation are shared, with Yiming coordinating electrical bring-up and Chris supporting fabrication. Each lead will coordinate with another member during laboratory testing; joint reviews and demonstrations involve the full team.

Table 3 Proposed Weekly Schedule

Week in 2026	Planned tasks and deliverable	Lead
--------------	-------------------------------	------

Sep 28–Oct 4	Finish PCB design and pass the design review before order; complete R&V and design document; confirm available parts and unresolved specifications. Course design document due Oct 2.	Team
Oct 5–11	Address design-review feedback; test Pi–ESP32 communication and one servo; prepare the breadboard demonstration and review the revised PCB. Design Review Oct 5–7; second PCB audit deadline Oct 8.	Yiming and Qianyu; Chris supports
Oct 12–18	Complete Breadboard Demo Oct 12–14; revise interfaces and protection design from bench results. Third PCB audit deadline Oct 15; final machine-shop revisions Oct 16 if applicable.	Team; Yiming and Chris lead revisions
Oct 19–25	Assemble and bring up the PCB when delivered; test power branches separately. Assemble the arm and establish camera and joint calibration. Maintain a bench setup if boards are delayed.	Yiming and Chris; Qianyu calibrates
Oct 26–Nov 1	Integrate perception, UART commands, servo control, and lighting; test the arm at restricted speed and reach. Use Oct 29 PCB audit round if a redesign is necessary.	Team; Qianyu and Yiming lead interfaces
Nov 2–8	Complete Progress Demo Nov 2–4 and individual reports Nov 4; record failures and revise software or hardware. Obtain repeatable end-to-end operation.	Team
Nov 9–15	Run the quantitative R&V procedures, including target accuracy, response time, simultaneous loading, and fault handling. Retain logs and repeat failed tests after fixes.	Qianyu, Chris, and Yiming by subsystem
Nov 16–22	Complete Mock Demo during the weekly TA meeting and Mock Presentation Nov 19–20; address feedback and repeat affected tests.	Team
Nov 23–29	Fall Break. Consolidate results and prepare presentation and report materials. Avoid relying on fabrication deliveries or laboratory access during the break.	Team, remotely as available
Nov 30–Dec 6	Complete Final Demo Nov 30–Dec 2 and Final Presentation Dec 3–4; submit video assignment Dec 4. Record the demonstrated configuration and results.	Team

Dec 7–11	Submit the final paper Dec 9; complete laboratory checkout and notebook submission Dec 10. Document remaining limitations and reusable hardware.	Team
----------	--	------

PCB manufacture and mechanical fabrication are dependencies for full assembly. We will continue software and interface testing on the bench while waiting for deliveries. The schedule reserves time for correction and repeated verification before the final demonstration. Exact team appointments and any course calendar changes must be checked with the TA.

4. Conclusion

4.1 Societal Impact

The robotic desk lamp is intended to reduce the effort required to reposition a light during writing, drawing, assembly, or other tasks that occupy a user's hands. A camera observes a single-hand interaction, and the controller moves the lamp toward the selected work area. This may benefit users who find repeated manual adjustment inconvenient. The prototype's usefulness will depend on reliable interaction and safe motion; we have not established a medical benefit or suitability for users with a particular disability.

The design also creates costs and environmental tradeoffs. Four servos, a camera, and two computing devices add expense, power consumption, and maintenance compared with a conventional lamp. We plan to measure power during operation and idle periods and to avoid unnecessary repeated movements. Replaceable assemblies and documented connections will allow damaged parts to be repaired individually. Reusable electronics will be retained, and discarded electronics will be handled through appropriate recycling. Affordability claims will depend on the verified project cost.

Gesture recognition can behave differently under varying lighting, backgrounds, hand appearances, and movement abilities. We will evaluate the chosen interaction under documented conditions and explain its limitations. Instructions should make the required gesture understandable without relying on spoken English alone. Broader accessibility and deployment in other settings will require further evaluation beyond the desktop prototype.

4.2 Ethical Considerations

Camera privacy is the primary data concern. The planned implementation processes images locally on the Raspberry Pi and uses only the information needed to interpret the interaction. Normal operation is intended to avoid persistent image recording and cloud upload. Before demonstrations, we will inform participants that a camera is active and limit its view to the interaction workspace. Any development recordings will require participant permission, a defined purpose, restricted access, and deletion after that purpose is complete. These measures address ACM Principle 1.6 on privacy [7].

An incorrectly recognized target can cause unexpected motion. ACM Principles 1.2 and 2.5 call for avoiding harm and evaluating system risks [7]; IEEE Code of Ethics commitments 1 and 5 address public safety and honest technical claims [8]. We will use these obligations to guide bounded motion and fault handling. Reports will identify test conditions, failed trials, and unverified functions rather than presenting a successful demonstration as proof of unrestricted reliability. Borrowed algorithms, libraries, and design material will be credited.

4.3 Engineering Standards

IEC 60598-1:2024 provides general safety requirements and tests for luminaires [9]. It is a relevant reference when reviewing the lamp's construction, wiring, and thermal safety. IEC 62471:2006 addresses photobiological safety of lamps and lamp systems [10] and is relevant when selecting the LED and reviewing exposure to its light. Exact applicable provisions and any additional luminaire-specific standard will need to be determined for the final assembly. We will use available supplier safety information and discuss the evaluation scope with the TA. No IEC certification or complete standards compliance is claimed for this prototype.

4.4 Electrical Thermal and Mechanical Safety

The planned system receives 12 V DC from an enclosed external AC/DC adapter, keeping mains wiring outside the custom PCB. High DC current can still overheat conductors, connectors, and components. After the supply and lamp specifications are finalized, we will select protection devices and wiring to match measured operating loads and foreseeable faults. Wiring will be insulated and secured against pulling and abrasion. Power branches will be tested individually using a current-limited bench supply before combined loading. We will measure supply behavior and the temperatures of the lamp, regulators, connectors, and wiring during sustained operation, and correct excessive heating before demonstration.

The arm presents pinch, collision, and tipping hazards. We plan to secure the base, restrict the reachable workspace, and impose joint, speed, and acceleration limits appropriate to the assembled arm. Tests will begin with individual joints at low speed, with hands and loose items outside the motion area. Cables will be routed to avoid entanglement. Joint limits and a stable base reduce particular risks but do not constitute obstacle detection. Operation will remain supervised until the mechanical range and fault behavior have been verified.

The ESP32 firmware will be designed to reject malformed, stale, and unreachable commands. On loss of the perception stream or communication, it should inhibit new motion and enter a tested stationary state. Whether the servos hold their position or are disabled must be decided from the arm's mechanical behavior; removing torque could allow the arm to fall. For supervised bench tests, an operator will have an accessible means of disconnecting power, with the arm mechanically supported where loss of power could cause a drop. We will test communication loss, reboot, and power restoration before allowing automatic motion to resume. Numerical limits and timeout values belong in the subsystem requirements and remain to be selected.

All members will follow the ECE 445 laboratory safety rules, including required training and having at least two people present in the laboratory [5]. Power will be disconnected before wiring changes, soldering, or mechanical adjustment. The move to four 12 V serial-bus servos

makes simultaneous loading, bus-command faults, and unintended motion explicit design concerns. Each protection measure must be verified on the current implementation before it is reported as effective.

References

- [1] Espressif Systems, “ESP32-S3-WROOM-1 & ESP32-S3-WROOM-1U Datasheet,” ver. 1.8, https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_data_sheet_en.pdf
- [2] Waveshare, “ST3215 Servo,” product specifications, <https://www.waveshare.com/product/st3215-servo.html>
- [3] Raspberry Pi Ltd., “Raspberry Pi hardware,” Raspberry Pi Documentation, <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>
- [4] Alpha and Omega Semiconductor, “AO3400A: 30V N-Channel MOSFET,” datasheet, rev. 3.1, Jul. 2023, <https://www.aosmd.com/sites/default/files/res/datasheets/AO3400A.pdf>
- [5] University of Illinois Urbana-Champaign, “ECE 445 course documentation,” Fall 2026, [Design Document](#); [Tolerance Analysis](#); [Calendar](#); [Safety](#)
- [6] Texas Instruments, “TPS56623x: 3-V to 18-V Input, 6-A Synchronous Step-Down Voltage Regulator,” datasheet, rev. B, Dec. 2023, <https://www.ti.com/lit/ds/symlink/tps566231.pdf>
- [7] Association for Computing Machinery, “ACM Code of Ethics and Professional Conduct,” 2018, <https://www.acm.org/code-of-ethics>
- [8] IEEE, “IEEE Code of Ethics,” revised Jun. 2020, <https://www.ieee.org/about/corporate/governance/p7-8.html>
- [9] International Electrotechnical Commission, “IEC 60598-1:2024: Luminaires — Part 1: General requirements and tests,” 2024, <https://webstore.iec.ch/en/publication/66620>
- [10] International Electrotechnical Commission, “IEC 62471:2006: Photobiological safety of lamps and lamp systems,” 2006, <https://webstore.iec.ch/en/publication/7076>