ECE 445

Spring 2024

Design Document

An Auto-Hand Chasing Lamp With Hand Gesture Control

Team 49
Feiyang Liu (feiyang5)
Yiyan Zhang (yiyanz3)
Jincheng Yu (jy54)

TA: Luoyan Li
February 22th, 2024

1. Introduction

Problem:

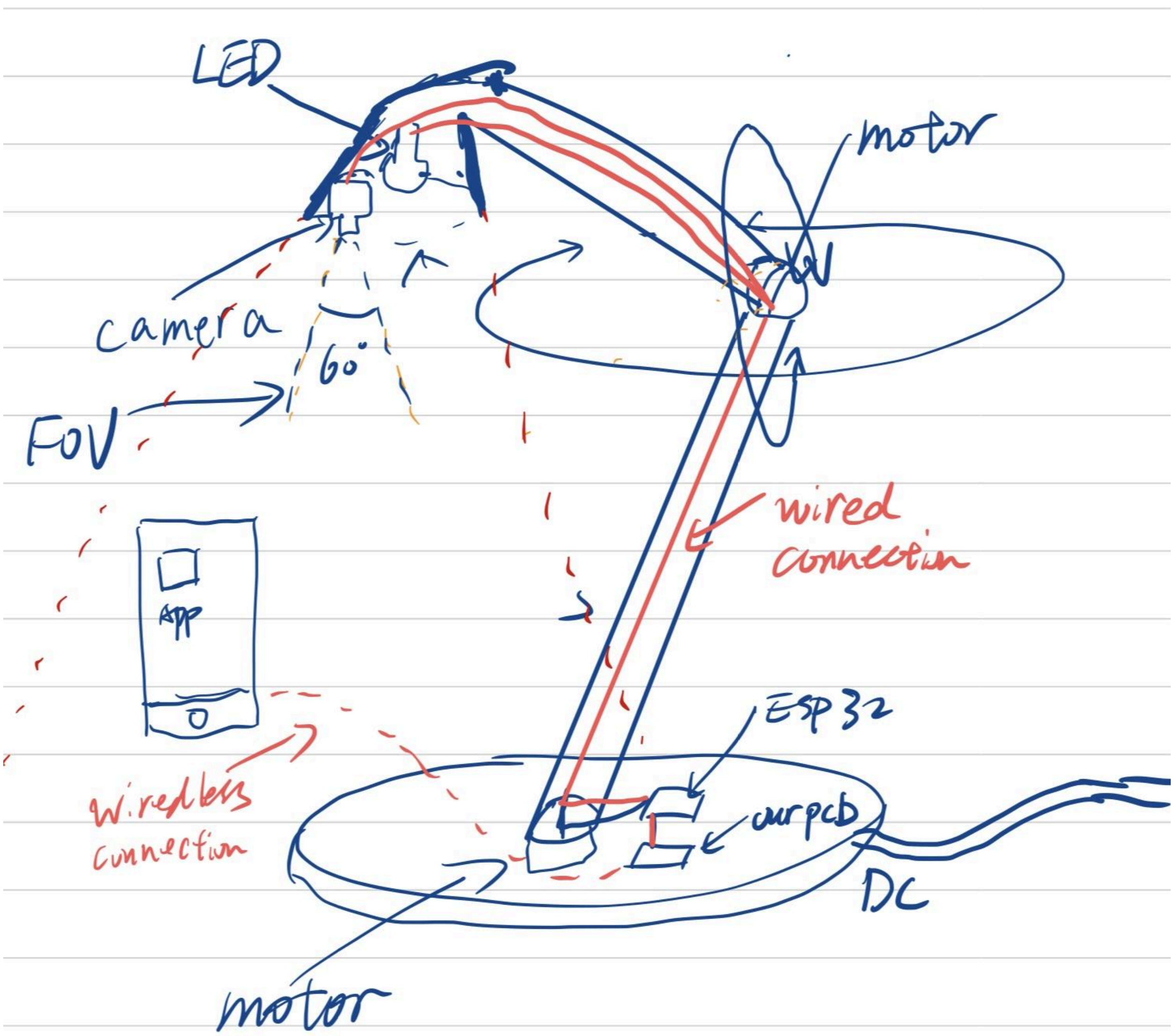
In industrial workspaces, inadequate lighting, especially from shadows, hinders precision tasks like soldering, impacting work quality and efficiency. Traditional lighting lacks flexibility and fails to meet specific task needs, leading to a demand for a smart, adaptable desk lamp.

Solution:

Our design is a desk lamp equipped with the capability to autonomously track the user's hand movements, thereby eliminating the common issue of shadows obscuring the work area. This is achieved through a combination of a camera, several servo motors, and a flexible mechanical arm, all of which work in tandem to reposition the lamp in real-time

The system utilizes gesture recognition technology, powered by an ESP32 module, to interpret hand gestures. This allows users to effortlessly control the lamp's functionalities—ranging from adjusting brightness and color temperature to switching the lamp on and off—without the need to physically interact with the device. Furthermore, the lamp is capable of executing more advanced commands, such as controlling external devices like computers to perform actions including playing music or adjusting volume. To complement its primary features, the lamp includes a customizable notification system that uses subtle changes in color temperature to alert users about important tasks or reminders, aiding in productivity and focus. Our smart desk lamp not only solves the problem of inadequate lighting but also introduces a level of interaction and functionality.

1.3 Visual Aid



1.4 High-level requirements list:

i. **Autonomous Tracking Accuracy:** The smart desk lamp must possess the ability to track the user's hand movements with a minimum accuracy(definition of accuracy: successfully recognize hands over other things like pen) of 95%, ensuring that the work area is consistently well-illuminated without manual adjustment of the lamp's position.

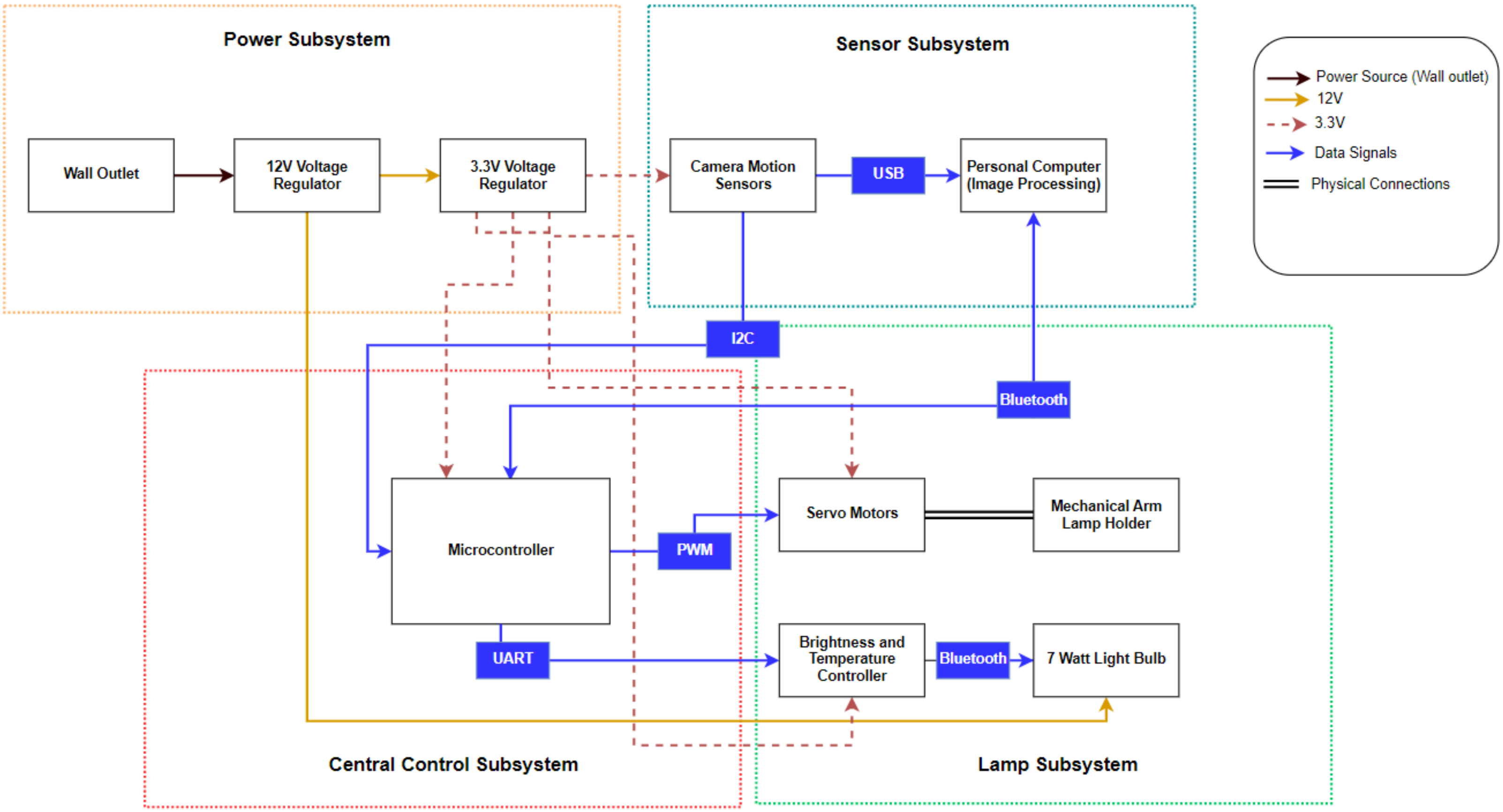
ii. **Gesture Recognition Responsiveness:** The system must be able to interpret and respond to predefined hand gestures within a timeframe of no more than two seconds from the moment of gesture completion. Hand gestures may include:

1. Swiping up or down to adjust the lamp's brightness.
2. Swiping right or left to change the color temperature.
3. A circular motion to turn the lamp on or off.
4. A pinching motion to activate focus mode, which reduces ambient lighting and increases task lighting.
5. A two-finger tap to toggle between preset lighting modes or scenes.

iii. **Light Adjustment Range:** The lamp must be capable of adjusting its brightness from a minimum of 100 lumens to a maximum of at least 1000 lumens and its color temperature from 2700K (warm light) to 6500K (daylight), to accommodate various lighting conditions and user preferences. Additionally, the lamp should offer a variable beam spread from a focused 15 degrees for precision tasks to a broad 120 degrees for general lighting, ensuring versatility across different types of workspaces and activities.

2. Design

2.1 Block Diagram



Auto-Hand Chasing Lamp's Block Diagram

2.2 Subsystem Overview

2.2.1 Sensor Subsystem

The sensor system is used to detect hand movement and upload the image to a computer to perform image processing trainings. It uses a ESP32 camera sensor to capture image and upload the image to an image processing platform on personal computer. The sensor sends digital output to the microcontroller when hand movements are detected, and the image processing platform would distinguish certain hand gestures through data processing.

2.2.2 Central Control Subsystem

This system contains a microcontroller that integrates the ESP32 module and necessary I/O modules. It needs to process images captured by the camera, determine how much each motor in the mechanical arm should move to track the bulb and be sensitive to specific gestures to adjust various parameters of the bulb. It can also communicate remotely with a computer to control specific programs and execute a trained AI gesture model. The lamp subsystem will receive outputs from microcontroller and change mechanical arm movement and light bulb activities based on the detected hand gestures. Detectable gestures will include fist and raising fingers from 1 to 5. The six gestures will be able to turn on/off, increase/decrease brightness, and increase/decrease light temperature accordingly.

2.2.3 Lamp Subsystem

The lamp subsystem makes up the physical lamp that will be used to track the user's hand movement. It's contains two key parts, which are a mechanical arm lamp holder and a light bulb. Three servo motors and a linear potentiometer are used to control the rotation and location of the lamp. The lighting bulb should be adjustable in terms of color temperature and brightness through instructions from the central control system.

2.2.4 Power Subsystem

The power subsystem provides the power needed for the sensors, microcontroller, and lamp system. We use a 12V voltage adapter to power the core lamp system. A separated 3.3V voltage adapter will be used to power smaller components including the camera sensor and microcontroller. The power need to sufficient to support the three servo motors in moving the physical mechanical arms, and a wall outlet would be used as the primary energy source.

2.3 Subsystem Requirements

2.3.1 Sensor Subsystem

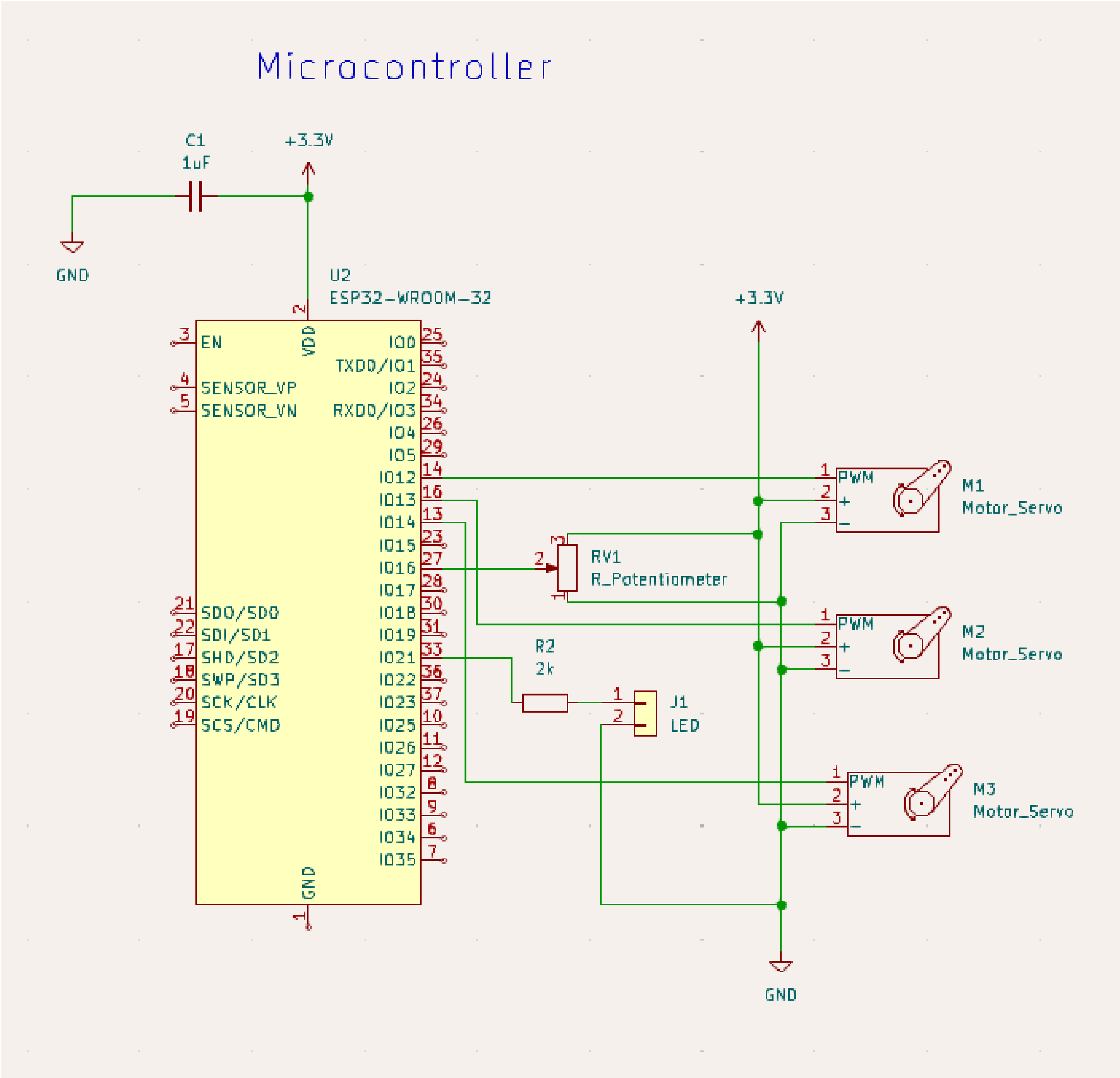
A sensor and an image processing platform will be used to in the sensor subsystem. The camera sensor, ESP32-CAM, can accurately capture image and communicate with a chosen image processing platform to track human hand movements. OpenCV or similar image processing tool will be used to distinguish human hands from other objects. This also ensures that we can identify certain hand gestures like pinching that are used to control the lamp.

Requirement	Verification

1. Able to capture image when sensor is powered on by 3.3V +/- 5%	● Set up ESP32 Cam on Arduino IDE and connect it to a 3.3V power source ● Use oscilloscope on ESP32 Cam power input to verify that power input is within the range 3.3V +/- 5% ● Check on personal laptop if Arduino IDE can receive images from ESP32 sensor
2. Upload readable camera image that could be used for image processing	● Set up OpenCV for image processing trainings ● Import images taken by ESP32 and verify if the images can be trained through OpenCV
3. Identify human hands with an accuracy > 90%, accuracy is measured by the tags assigned to the taken images during training process and calculated through the formula (images tagged with human hands / all human hands images taken)	● Set up OpenCV for image processing trainings ● Use ESP32 Cam to take 50 hands images and perform trainings through OpenCV ● Check all trained images' properties to verify if at least 45 images are tagged with human hands
4. Able to accurately count stretched out fingers within the range 0-5	● Set up OpenCV for image processing trainings ● Use ESP32 Cam to take 20 hands images with stretched out fingers ranging from 0 - 5 and perform trainings through OpenCV ● Check all trained images' properties to verify if the images are correctly tagged with finger number within the range 0 - 5

2.3.2 Central Control Subsystem

A selected microcontroller, ESP-WROOM-32 Microcontroller, will be able to process captured image from ESP32 sensor and connect with the lamp system to determine mechanical arm movement and lamp lighting.



Microcontroller's Connection to Lamp Subsystem

Microcontroller's outputs will be connected to the lamp subsystems. A linear potentiometer and three servo motors will be used to move mechanical arms based on signals received from ESP32 microcontroller.

Requirement	Verification

1. Able to communicate data through I2C, PWM and UART protocols	● Set up and program the ESP-WROOM-32 Microcontroller through Arudino IDE ● Connect oscilloscope between the ports used to transfer I2C, PWM and UART data to verify if signals are sent to sensor and lamp subsystems
2. Communicate with the mechanical arm to follow human hands with a delay < 0.5s	● Set up and assemble the complete control, lamp, sensor, and power systems ● Initiate hand movement in front of ESP32 Cam ● Use timer app on mobile phone to ensure the reaction speed delay of mechanical arm tracking is below 0.5s
3. Communicate with the light controller to switch on/off and adjust brightness with a delay < 0.5s	● Set up and assemble the complete control, lamp, sensor, and power systems ● Stretch out one finger (turn on light gesture) in front of ESP32 Cam ● Use timer app on mobile phone to ensure the light switching delay is below 0.5s

2.3.3 Lamp Subsystem

The lamp system ensures the movement of the light bulb and controllable lighting. The lamp uses a mechanical arm to move the light bulb in order to track hand movement. Two SF3218MG servo motors, one FS90R servo motor, and a P0915N potentiometer are used to control the lamp movement. For the base and important joints, we will use the SF3218MG with higher toruqe for high pressure steering. A 7 Watt light bulb such as Govee A19 with adjustable temperature and dimming will be used to control the lamp lighting. The light bulb has built in traic that could control supplied energy to limit brightness level and respond to microcontroller. By taking inputs from the Microcontroller, the light controller will select light mode and adjust brightness based on the user's hand gestures.

Requirement	Verification
1. Servo motors are able to move the mechanical arm when powered on by 3.3V +/- 5%	● Set up servo motors and mechanical arms and connect servo motors to 3.3V power source ● Activate SF3218MG and FS90R motors and observe if mechanical arms can be moved by motors
2. Mechanical arm can move in x, y, and z directions	● Set up servo motors and mechanical arms and connect servo motors to 3.3V power source ● Activate each of the three servo motors corresponding to one of the three movement dimensions, observe if each of the servo motor can move the mechanical arm in the corresponding to x, y, and z directions
3. Light bulb can adjust brightness and temperature when powered on by 12V +/- 5%	● Assmeble the complete power, sensor, central control, and lamp subsystems and connect light bulb to a 12V power source ● Stretch out 2-3 fingers in front of ESP32 Cam to verify if light bulb can increase/decrease brightness through dimming ● Stretch out 4-5 fingers in front of ESP32 Cam to verify if light bulb can change color temperature

2.3.4 Power Subsystem

The power system support the energy needed for the lamp, sensor, and central control subsystems. A wall outlet will be the primary energy source. Extra 12V and 3.3V voltage adapters are used to power up separate components in the design.

Requirement	Verification

1. Wall outlet supports up to 2A current draw	● Use breadboard and resistors to build a testload that could draw 2A current ● Connect wall outlet to test load and use use multimeter in between the test load to verify if it could read from 0A to 2A currrent
2. Primary voltage regulator enables 12V +/- 5% voltage output	● Connect 12V voltage regulator to a wall outlet ● Use oscilloscope on the voltage regulator output to verify if it supports 12V +/- 5% output
3. Secondary voltage regulator enables 3.3V +/- 5% voltage output	● Connect 3.3V voltage regulator to a wall outlet ● Use oscilloscope on the voltage regulator output to verify if it supports 3.3V +/- 5% output

2.4 Tolerance Analysis

In our design, we will be utilizing the ESP32 controller, which has a minimum load requirement of 0.5A. Additionally, we will also be using several FS90R motors to control the robotic arm, with a stall current of 800mA. If the FS90R can't do the task we need correctly, we'll use the SF3218MG, a motor with 20kg of high torque that will meet our needs.

$$\tau = F \cdot r \cdot \sin\theta$$

Referring to the torque formula, the entire lamp would weigh at most 2kg. Assuming a rotation radius of 20cm and angle of 270 degree. We calculate that the minimum torque required to move the lamp would be $\tau = 9.81*2*0.2*\sin(90) = 3.5$ N-M. SF3218MG motor could support torque greater than 5 N-M and is expected to meet our needs.

We have also planned to use a power supply capable of providing 2A of current, so we can safely utilize this power source. Furthermore, we need to ensure that the cameras' field of view (FOV) can capture as much of the scene as possible to avoid losing track. For the ESP32-CAM, we are using the OV2640, which offers around 65 degrees of FOV, sufficient to capture adequate scene information. Additionally, the maximum resolution can reach 1622x1200. When used separately from a computer, the ESP32-S3 is capable of shooting at 400 × 300 pixels and running relatively small AI models at the same time. This gives us enough computing power and pixels to find the hand bones.

3 Cost Analysis

The cost analysis for this project is primarily divided into two main components: labor costs and parts costs. We have estimated the hourly wage for each project partner at $56, based on market research for graduates from Electrical and Computer Engineering (ECE) at the University of Illinois. Below is a detailed cost analysis:

3.1 Labour Costs

- The assumed hourly wage for each partner is $56.
- The total hours each person should use for the project is 20hours/person, And we assume to finish the project in 8 weeks.
-
- The formula for total labor cost is: `56 x 2.5 x 20 x 3 x 8 = 67200`.

3.2 Parts Cost

The following table lists all the parts required for the project, including description, manufacturer, part number, quantity, and cost, along with estimated machine shop labor hours needed to complete the project.

Item	Manufacture	Part Number	Quantity	Unit Price(USD)	Total Price(USD)
ESP32-S3-CAM	Freenove	-	1	20	20
SF3218MG Metal Gear Digital Servo	Sunfounder	-	2	20	40
FEETECH FS90R (set of five)	FEETECH	-	1	26	26
P0915N	TT Electronics	-	3	8	24
ESP-WROOM-32	HiLetgo	-	1	10	10
Total					120

3.3 Grand total

The grand total will include the sum of the labor costs and the parts costs. The specific labor cost will need to be calculated based on the actual hours worked by each partner. The total parts cost is $110. All labor costs amounted to $67,200.At the same time, we also need to consider the expenditure of PCB board design completion. This will be updated after we decide on the design, tentatively set at $20.
The total cost is therefore approximately 67,400.

2. Schedule:

Week 1 (Feb 22 - Feb 28): Initial Planning and Architecture

Task: Select architecture and finalize project scope.

Members Involved: Jincheng (Lead), Yiyan, Feiyang.

Deliverables: Project scope and architecture plan

Week 2 (Mar 1 - Mar 7): Detailed Design Phase

Task: Design sensor subsystem; Yiyan begins ESP32-CAM integration.

Members Involved: Feiyang (Lead on sensor design), Yiyan (ESP32-CAM), Jincheng.

Deliverables: Sensor subsystem design, initial ESP32-CAM setup.

Week 3 (Mar 8 - Mar 14): Control and Lamp Design

Task: Design central control subsystem; start lamp subsystem design.

Members Involved: Jincheng (Lead on central control), Feiyang (Lead on lamp design), Yiyan.

Deliverables: Control and lamp subsystem designs.

Week 4 (Mar 15 - Mar 21): Parts Procurement and Testing

Task: Order and test individual components; finalize ESP32-CAM integration.

Members Involved: Yiyan (Lead on ESP32 and testing), Jincheng, Feiyang.

Deliverables: Components ordered, initial component tests, ESP32 integration.

Week 5 (Mar 22 - Mar 28): Assembly of Subsystems

Task: Assemble sensor and central control subsystems.

Members Involved: Feiyang (Lead on assembly), Jincheng, Yiyan.

Deliverables: Assembled sensor and control subsystems.

Week 6 (Mar 29 - Apr 4): Lamp Assembly and System Integration

Task: Assemble lamp subsystem and integrate with control system.

Members Involved: Jincheng (Lead on lamp assembly and integration), Feiyang, Yiyan.

Deliverables: Lamp assembly, integrated system.

Week 7 (Apr 5 - Apr 11): System Refinement and Demo Preparation

Task: Refine integrated system and prepare for mock demo.

Members Involved: Feiyang (Lead on refinement), Jincheng, Yiyan.

Deliverables: Refined system, prepared demo.

Week 8 (Apr 12 - Apr 15): Final Testing and Demo Execution

Task: Conduct final system testing and execute mock demo.

Members Involved: Yiyan (Lead on final testing), Jincheng, Feiyang.

Deliverables: Final testing completion, successful mock demo.

4 Ethics and Safety

In terms of ethics and morality, our team adheres to the IEEE Code of Ethics adopted by the IEEE Board of Directors in June 2020. We firmly believe that as members of UIUC, a world-renowned university, we need to hold ourselves to the highest academic and ethical standards. We hope to change the world with our technology. Therefore, when working in our team, we will take the following measures, including but not limited to:

> 1: Diligently Studying Technology and Actively Communicating with Guides (TAs, Professors): Within our team, we strive to learn as much technology as possible in this project, as well as how to successfully combine the knowledge we have learned into a viable project. We will seek various resources, including but not limited to the internet, videos, and books, and actively seek advice from professors with insights in this field. At the same time, we will also actively share and exchange experiences and insights within the group to help everyone gain sufficient knowledge and skills. In this task, we will learn about relatively cutting-edge fields such as web development, AI vision, firmware development, PCB research and development, and robotics. Perhaps our ability to delve deeply is temporarily limited, but at least we can explore these fields and gain insights. We will continue to learn and apply this knowledge in the future in academia or the industry.
>
> 2: Developing Comprehensive Feedback and Testing Plans: In our team, because there are many different areas of content, we will develop comprehensive testing plans for subsystems including robotic arms and AI, and timely feedback these test results to TAs while planning the next strategy.
>
> 3:Treating Everyone with Respect and Kindness and Ensuring These Codes Are Adhered To: To ensure good teamwork and communication, we have set up chat groups, a GitHub repository, and Google Drive space to share resources. We ensure all technical details can be tracked.

Laboratory Usage and Safety

Adherence to Laboratory Rules: Strictly following all laboratory rules and guidelines is essential for ensuring safety. This includes wearing appropriate personal protective equipment (PPE), such as safety goggles, gloves, and lab coats, when necessary.

Supervised Mechanical Work: When developing or working on mechanical parts, supervision or collaboration is crucial. Working alone on mechanical components can increase the risk of accidents, so we'll ensure that team members are present or that a lab supervisor is available for assistance.

Equipment Training: Before using any lab equipment, team members will receive proper training. This training will cover the operation of the equipment, safety procedures, and emergency protocols.

Digital Security

Secure Wi-Fi and Bluetooth Connections: Given the project's reliance on Wi-Fi and Bluetooth for communication, ensuring these connections are secure is vital. We will implement encryption protocols such as WPA2 for Wi-Fi and LE Secure Connections for Bluetooth to protect against unauthorized access.

Data Privacy: Protecting the privacy of users interacting with our system is paramount. We will design the system to collect only the necessary data, ensuring it is stored securely and that users are informed about how their data is used.

Reference

IEEE code of Ethics. IEEE. (n.d.-a). https://www.ieee.org/about/corporate/governance/p7-8.html