Sit & Chow – Smart Training Dog Feeder

Final Presentation

Group 25

Advisor:

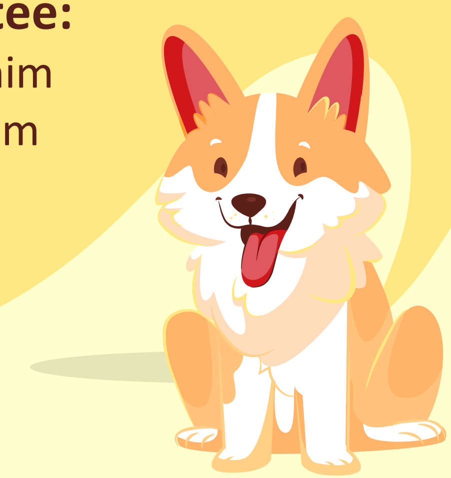
Dr. Saleem Sahawneh

Review Committee:

Dr. Shahana Ibrahim

Dr. Truong Nghiem

Dr. Tong Wu



Group 25 Members

Andres Arzola
Computer Engineering

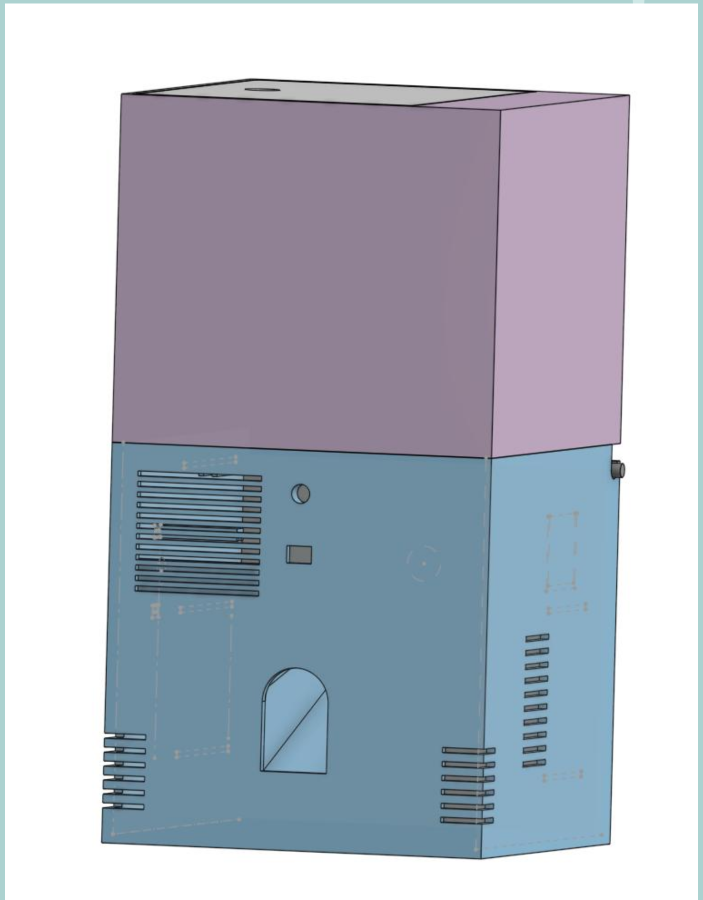
Andrew Balmes
Electrical Engineering

Natalie Nguyen
Electrical Engineering

Maria Tran
Computer Engineering

Motivation and Background

- Traditional automatic feeders only have the convenience of feeding pets consistently at scheduled times. However, these systems do not address behavioral training during feeding.
- **Problem:**
    - Some dog owners lack a reliable and consistent way to enforce consistent training during feeding time, especially when they are not physically present. As a result, dogs may develop undesirable behaviors, such as barking or jumping.
- **Our solution:**
    - The proposed system combines automatic feeding with behavioral training by requiring dogs to “sit” before food is dispensed by combining an automated feeder with machine learning.



Goals

1

Provide a reliable, automated feeding solution for dogs.

2

Reinforce obedience training by requiring a dog to sit before eating.

3

Allow for owners to conveniently manage feeding times and portions through a mobile app.

4

Monitor system status and system reliability.

5

Allow for owners to check in with their dog at any point of the day through the mobile app and camera.

Objectives

1

Develop a food dispensing system that operates on a timer.

2

Incorporate a camera to detect a dog’s posture.

3

Design a mobile app for scheduling feeding times, adjusts portion sizes, logs feeding history.

4

Include sensors to check food supply level and how far a dog is from the system.

5

Include a speaker on the system and add the feature on the app.

Engineering Specifications

Aspect	Specifications
Feeding Schedule Accuracy	Within 3 minutes
Low Food Storage Range	Over 20cm
Depth Sensing range	Up to 2 feet
Sound Output	60-80 dB
FPS	Up to 15fps
Posture Detection Accuracy	At least 70%
Mobile App Features	Creating and editing feeding schedules, quick feed, feeding logs, low food supply notifications, and user authentication
Data Logging Retention	Minimum 30 days
Food Storage Capacity	Maximum 5 kilograms

Hardware Comparisons

ESP32-S3

Why We Chose The ESP32-S3

- Native Camera Interface for OV5640
- More Memory For image Processing
- Able to handle multiple sensors
- Built in Wi-Fi and USB

	ESP32-WROOM-32	**ESP32-CAM**	**ESP32-S3**
Camera Support	None (no DVP/CSI interface)	Built-in OV2640 2MP camera (via GPIO)	Native DVP camera interface (supports OV2640, OV5640)
USB Inferface	Requires external UART adapter	Requires FTDI programmer	Native USB OTG — direct flashing and debugging
GPIO Availability	30+ usable GPIOs	6–7 available after camera/SD use	30+ usable GPIOs (depending on module configuration)

OV5640 Camera

OV5640

- High Resolution
- Reliable under varying lighting
- DVP interface with ESP32-S3
- 30 FPS @ 1080p for smooth Capturing

	OV2640	**OV5640**	**Realtek AMB82**
Resolution	2 MP (1600x1200)	5 MP (2592x1944)	2 MP (1920x1080)
Frame Rate	15-30 Fps	30 fps / 1080p	30 fps /1080p
Cost	Low	Moderate	High
Ecosystem Support	Large Ecosystem	Good Ecosystem	Less support than the other two cameras

Motor & Motor Driver

Motor	Pros	Cons	Notes
Stepper (NEMA-17)	Open-loop precision, easy microstepping, stall detect with modern drivers	Higher current draw and heat when holding, more expensive	Best choice for auger metering
DC + gearbox	Low cost, efficient	Requires encoder or feedback loop for precision	Suitable for timed dispensing systems
RC Servo	Simple control set up	Limited torque and shorter lifespan	Ideal for flap gates only

Feature	A4988	DRV8825	TMC2209
VMOT Range	8–35V.	8.2–45V.	Up to ~28V.
Max Current (typical use)	~1–2A/phase with cooling.	~1.5–2.2A/phase with cooling.	2A RMS, 2.8A peak/phase.
Max Microstepping	1/16 step.	1/32 step.	1/256 (via interpolation).
Noise Level	High (audible whine).	High–moderate.	Very low (StealthChop2).
Special Features	Basic protections.	Strong current regulation, protections.	StealthChop2, SpreadCycle, StallGuard4, CoolStep.
Suitability for Dog Feeder	OK for budget prototypes.	Overkill voltage, too noisy.	Best match: quiet, smart, enough current.

- Update: after testing and considering, we decided to switch to DRV8825

Speaker

Feature	1W 8Ω Speaker	5W 8Ω Speaker	3W 4Ω Full-range Speaker
Loudness (Sensitivity)	Less efficient and likely quieter than the 4Ω option for the same input voltage.	Less efficient than the 4Ω option.	A 4 Ω speaker can be louder than 8 Ω speaker
Pros	Low power rating means very modest current or drive required. 8Ω is a common safe impedance for small amplifiers (the microcontroller audio driver likely can handle it). Lower cost, smaller size.	Higher power handling given more headroom for louder notifications, which may help in presence of ambient noise. Still 8Ω, which is compatible with many drivers.	4 Ω impedance draw more current at a given voltage, meaning more available acoustic output for the same voltage – useful in low-voltage system. A full-range speaker covers the tone/frequency band of interest (the dispense chime) without needing separate tweeter/woofer 3W rating is the modest but sufficient for a small feeder environment.

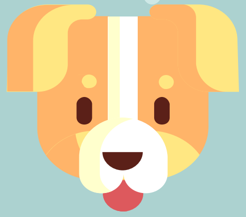
- We decided **3W 4Ω Speaker** is a best balance for what we need: moderate power (3W) suitable for embedded drive, lower impedance (4Ω) to use voltage efficiently, and full range for nice tone. The other options either under-power (1W) or over-spec (5W) potentially wasting cost and driving complexity.

Amplifier

Features	Class A	Class AB	Class D
Efficiency	~25% (very low).	~50-70% (moderate).	>90% (very high).
Heat Dissipation	High, requires a large heat sink.	Moderate.	Low, minimal or no heat sink needed.
Fidelity	Very high, low distortion.	High, with minimal crossover distortion.	High, modern designs offer excellent audio quality.
Advantage	Excellent linearity and low distortion (very little crossover distortion) because the device is always on and in its linear region. Good for high-fidelity audio (audiophile territory).	Good compromise: lower distortion than Class B, better efficiency than Class A. Widely used in home audio stereo amplifiers.	Very high efficiency because the devices are either fully on or off (hence minimal wasted power as heat) Compact size, lower heat sink requirement, good for embedded systems (battery or low-power).

Feature	MAX98357A	PAM8403	PAM8302
Input Type	I²S Digital	Analog	Analog / PWM
Channels	Mono	Stereo (2 ch)	Mono
Noise & Quality	Clean digital path. Low hiss. Great clarity.	Acceptable for simple tones. Noise depends on analog path.	Can pick up noise. Dependent on analog cleanliness.
Power Output (5V)	~3.2W into 4Ω	~3W ×2 into 4Ω	~2.5W into 4Ω
MCU Compatibility	Native support on ESP32 & similar MCUs	Works with PWM/DAC	Works with PWM or simple DAC
Ease of Use	Moderate. Needs I²S setup but straightforward.	Easy. Many ready boards.	Very easy. Minimal configuration.

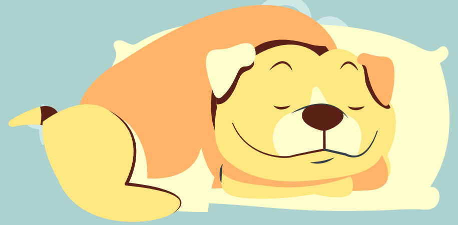
- We started to work with MAX98357A at first. However, after considering about soldering it on main board because of its size, we decided to switch to a new model which is LM386.



Programmer

Programmer	CP2102	FT232RL	CH340
Auto-flash	Easy	Easy	Works but less consistent
Driver issues	Rare	Rare	Sometimes
Cost	Low	High	Very low

- Reliable firmware flashing
- Better driver experience
- Cleaner hardware design
- Cheap

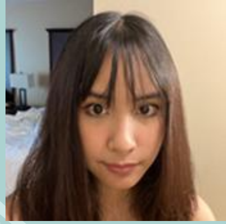


Sensing Storage Capacity

Feature	Load cell Sensor	Infrared Sensor (IR)	Capacitive Sensor	Ultrasonic Sensor
Environment Factors	Mostly immune, especially with hermetic sealing. Sensitive to vibrations if measuring minor	Prone to interference from dust, direct sunlight, and changing ambient light.	Sensitive to temperature, humidity and material build-up on the sensor.	Can be affected by extreme temperature changes, but generally reliable in typical home environments.
Pros	Measures actual weight for precise portion control; Can be very accurate and reliable for batching.	Inexpensive and easy to implement.	Low cost, durable, and suitable for non-conductive materials.	Non-contact method is hygienic and not affected by food type or color; Affordable, widely available, and simple to interface with microcontrollers.

Feature	RCWL-1601	JSN-SR04T	A02YYUW
Key Features	HC-SR04 protocol but works at 3.3 V or 5 V	Waterproof transducer on cable	Waterproof ultrasonic with digital UART output
Pros	Direct ESP32 compatibility; no level shifting; low cost; compact	Very rugged; moisture/dust resistant; long range	Waterproof; noise-filtered digital output; good accuracy
Cons	Not waterproof; dust may degrade accuracy over time	More expensive; mounting more complex; cable + puck footprint	More expensive; UART parsing required; slower update rate
Suitability for Dog Feeder	Best choice for indoor dog feeder; ideal electrical compatibility	Excellent for outdoor or washable feeders; overkill for indoor use	Good for industrial feeders; more complex than needed

- Since we only use it inside the house and check for how much food in storage, we decided RCWL-1601 is the best choice for indoor dog feeder and less complex than others.



Depth Sensor

- **High Distance Accuracy**: cm-level precision, great for verifying that the dog is within feeding range.
- **Digital I2C Interface**: clean integration with the ESP32-S3. No need for pulse timing measurements like ultrasonic sensors.
- **Quiet**: ToF sensor does not emit any audible sounds that may disturb the dog like ultrasonic waves.
- **Fur**: most sensors are sensitive to reflections and light. The ToF sensor is sensitive to sunlight, but the system must be operated indoors.

Sensor	Features
Time-of-Flight Sensor (ToF)	Works in indoor environments Easy to integrate with I2C connection

Power Management – Power Supply Type

Power Supply	Advantages	Disadvantages
Bench Power Supplies	Highly accurate voltage regulation. Adjustable current limiting prevents component damage. Ideal for debugging, prototyping, and electrical characterization.	Expensive compared to consumer power sources. Bulky and not portable. Not suitable as a final embedded power solution.
Wall Adapters (AC-DC adapters)	Reliable, continuous 24/7 power. Electrically isolated for safety. Available in many voltage/current ratings. Perfect for stationary appliances.	Requires access to a wall outlet. System loses power during outages unless battery backup is added. Cheap adapters may cause electromagnetic noise.
Battery Packs	No auto-shutoff issues (unlike USB banks). Flexible voltage design (1.2 V–12 V+). Inexpensive and safe. Excellent for backup power.	Heavy and bulky. Voltage decreases as cells discharge. Need periodic replacement or charging.
Switching Power Supplies (SMPS)	Very efficient (>85–95%). Minimal heat generation. Supports high input voltage ranges. Available as ICs or off-the-shelf modules.	Switching noise and EMI. Layout must follow strict rules. Requires additional components (inductor, diode, capacitors).

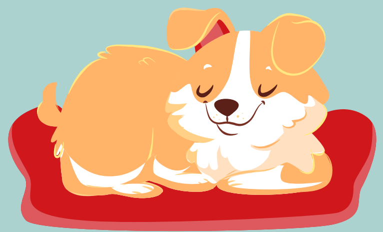
- Simple
- Maintain the product with enough power to provide continuous and smooth operation

Power Management – Wall Adapter

Feature	12V Wall-Mount Adapter	USB-C PD Charger + PD Sink	Desktop Brick
Complexity on PCB	Very low (just barrel jack & filtering)	High: PD sink IC, CC negotiation, etc.	Low (similar to wall-mount)
Cost & BOM Impact	Low total cost	Higher total (charger + PD sink BOM)	Slightly higher than wall wart
User Familiarity	Very familiar	Also familiar, but expects USB devices	Common for laptops/routers

Feature	Facmogu 12V 3A Adapter	Mean Well GS36U12-P1J (36 W)	CUI SMI36-12-V-Px (36 W)
Barrel Connector Fit	Flexible (two barrel sizes)	Specified barrel size	Specified barrel size, known connector
Cost / Availability	Likely lower cost	Slightly higher cost but trade-off in reliability	Slightly higher cost but trade-off in reliability

- Simple
- Cheap
- Easy to use



Power Management – Voltage Regulators

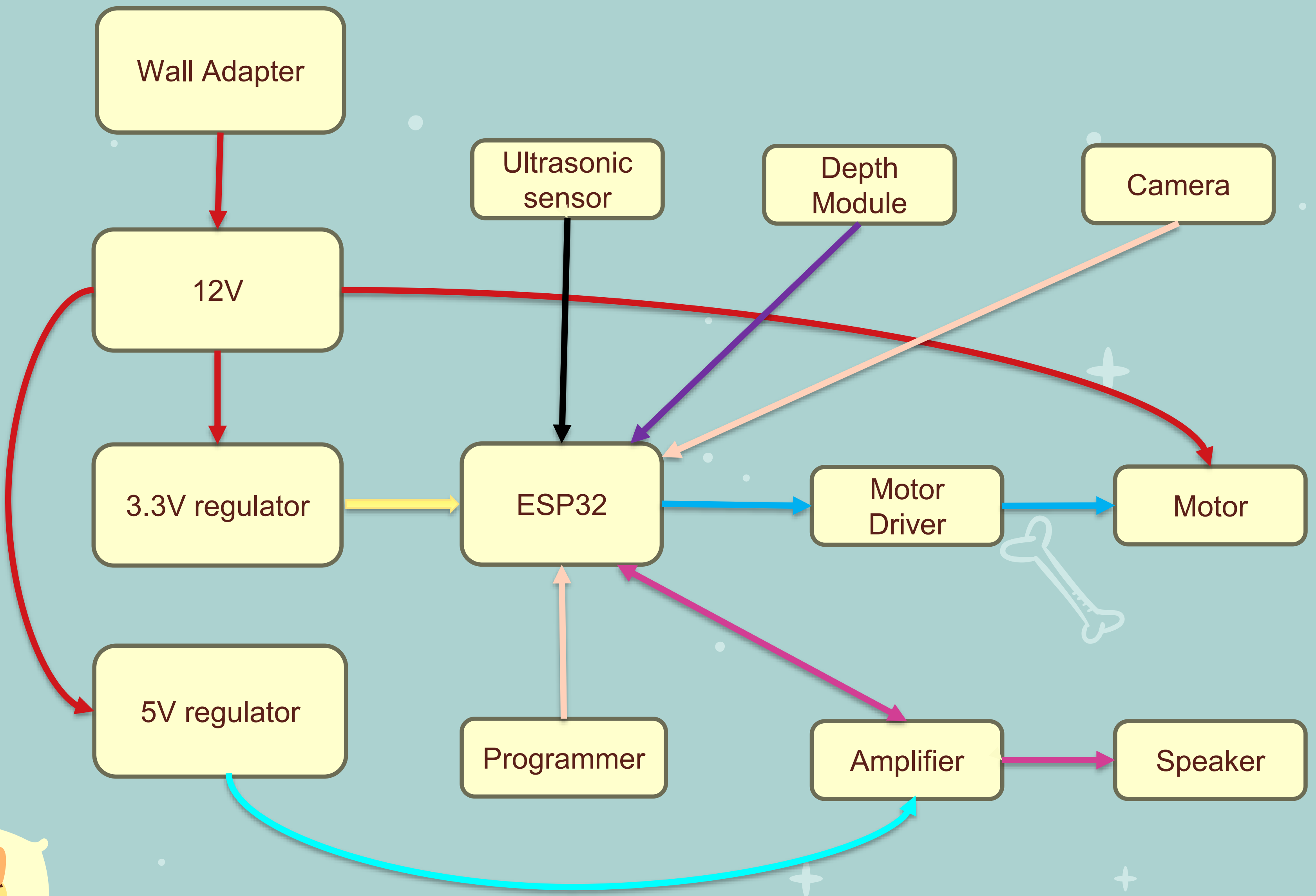
Feature	Linear Regulators	Non-Synchronous Buck Regulators	High-Current Simple Switcher Buck Regulators	Off-the-shelf Buck Modules
Typical Efficiency (12→5 V)	~42%.	~70–85 %.	~85–92 %.	Similar to IC used.
Current Capability	Up to a few A (with huge heatsink).	Up to ~3A.	Up to 5A.	Often 2–3A typical.
Thermal Behavior	Enormous heat: (12–5)·I, up to 35W at 5A.	Moderate heat, manageable at 2–3A.	Several watts of loss at 5A, heat sinkable.	Depends on module quality.
PCB Complexity	Very simple.	Moderate.	Moderate.	Very easy.

Feature	LM2596	LM2679	LM22678
Topology	Non-synchronous buck.	Non-synchronous buck.	Non-synchronous buck.
Vin range	4.5–40 V.	8–40 V.	4.5–42 V.
Vout option	Fixed 5V available.	Fixed 5V available.	Fixed 5V available.
Iout max	3A.	5A.	5A.
Typical efficiency (12→5 V)	~75–85% (design-dependent).	~85–92% (design-dependent).	~85–92% (design-dependent).
Layout difficulty	Low–moderate.	Moderate (comfortable for 2-layer).	Moderate–high (more EMI care).
Fit for 5V at 5A dog feeder	Undersized.	Excellent, recommended.	Excellent, but more layout-sensitive.

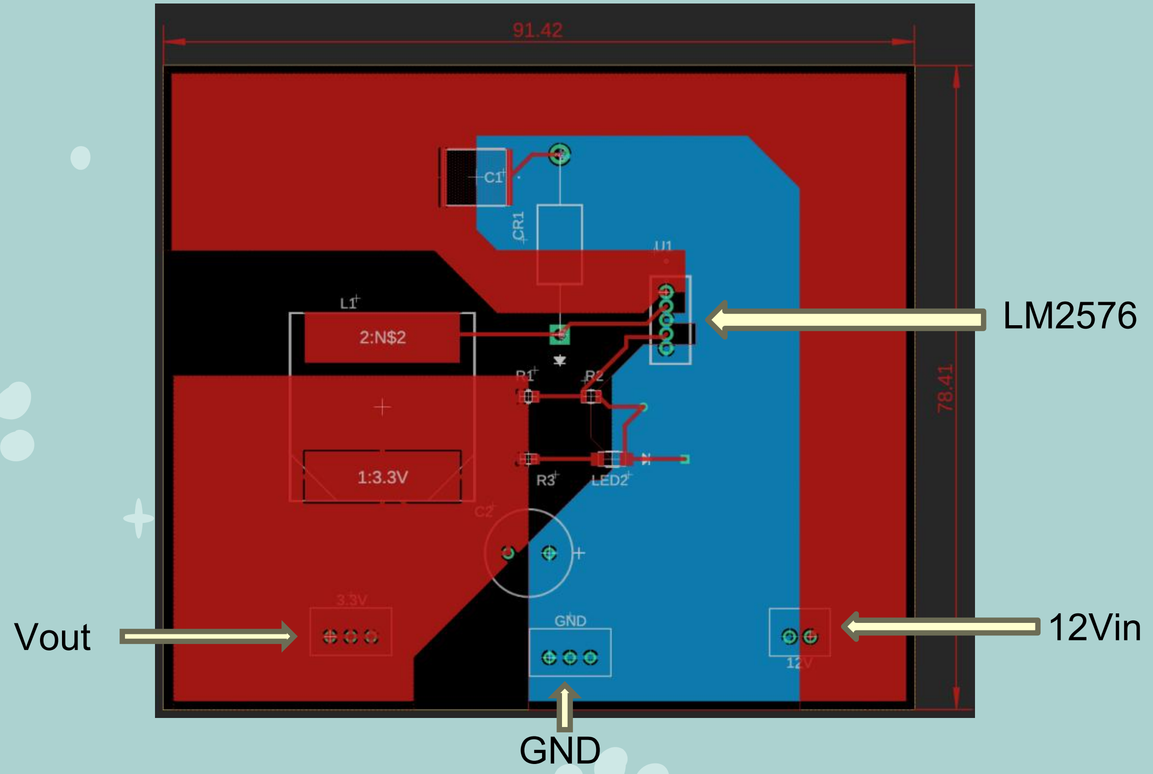
- Problem: SMD chip is too small and cannot test before doing PCB.
- Solution: Use LM2576-ADJ for both 3.3V and 5V.

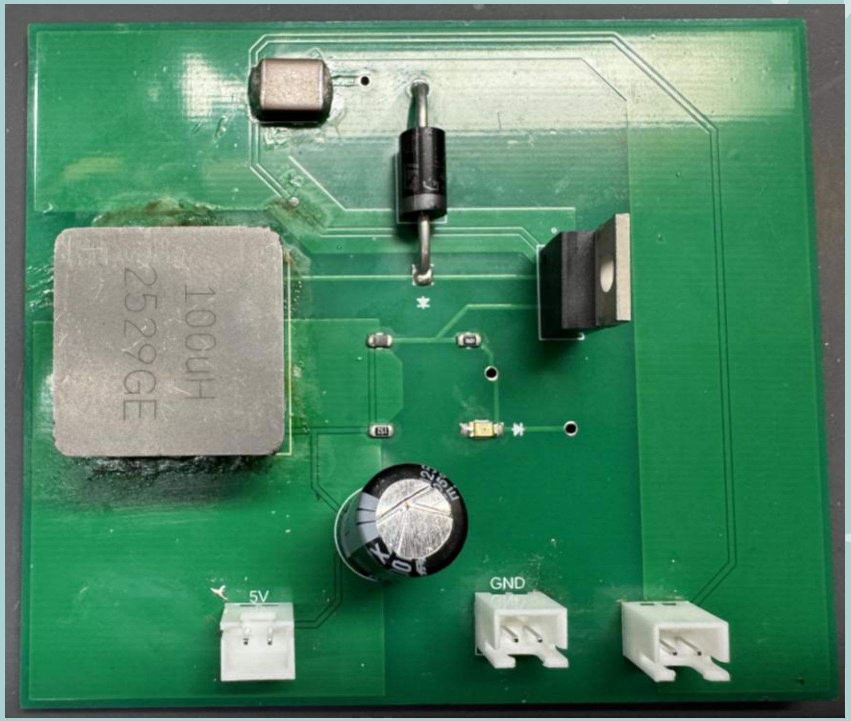
PCB Design

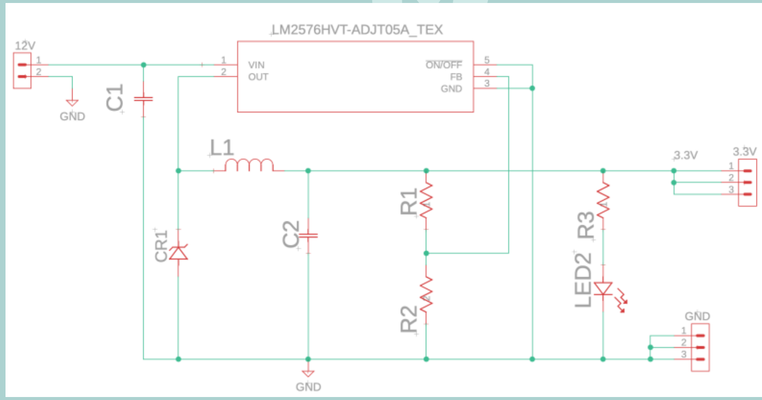
Hardware Design



Voltage Regulators

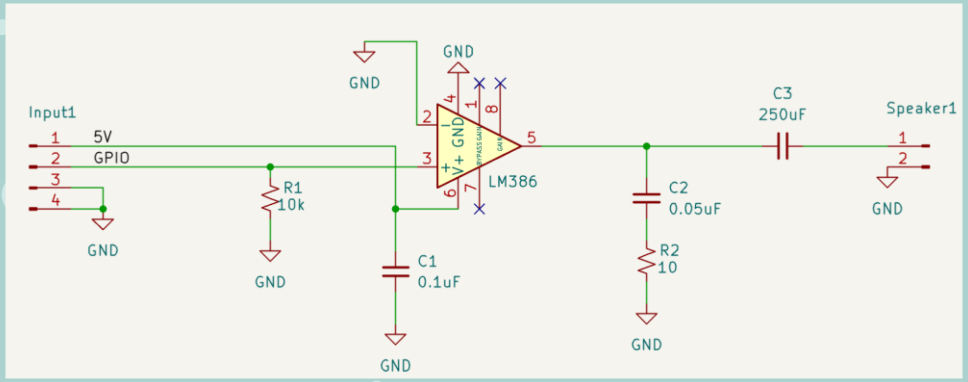


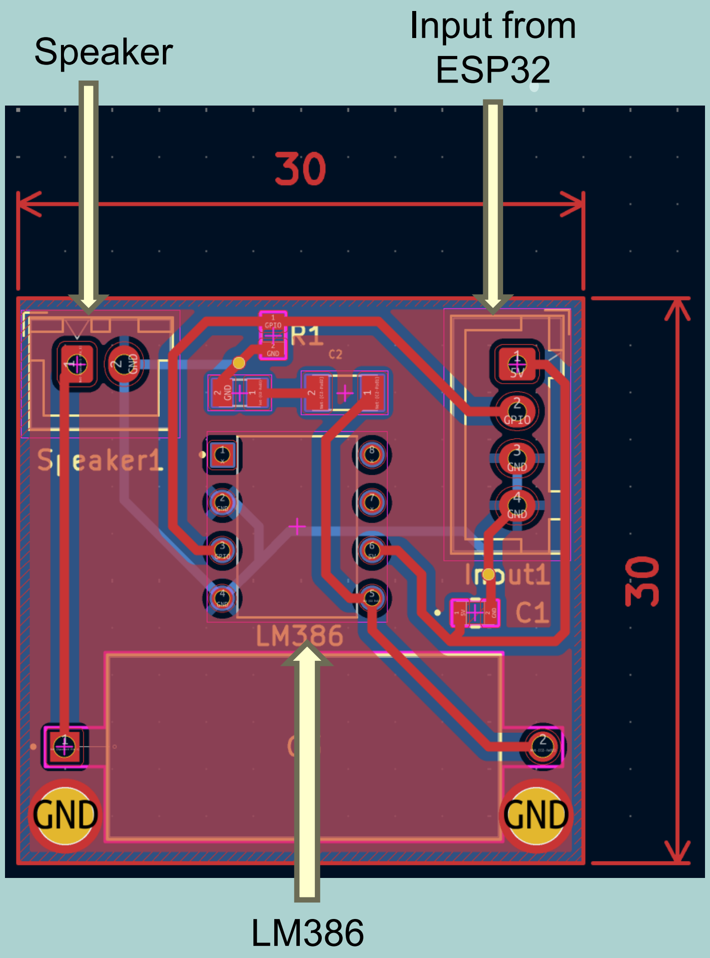


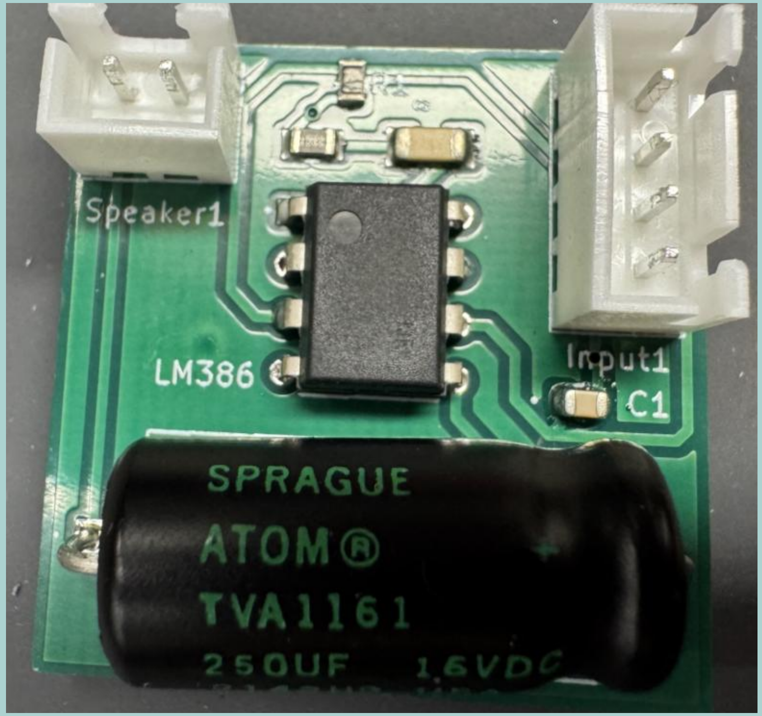


- Since using LM2576-ADJ, we can use this one for both 3.3V and 5V.
- We only have to adjust values of R1 and R2 to get the output we need.
- We also need to be careful to decide how much current we draw out for this.

Amplifier

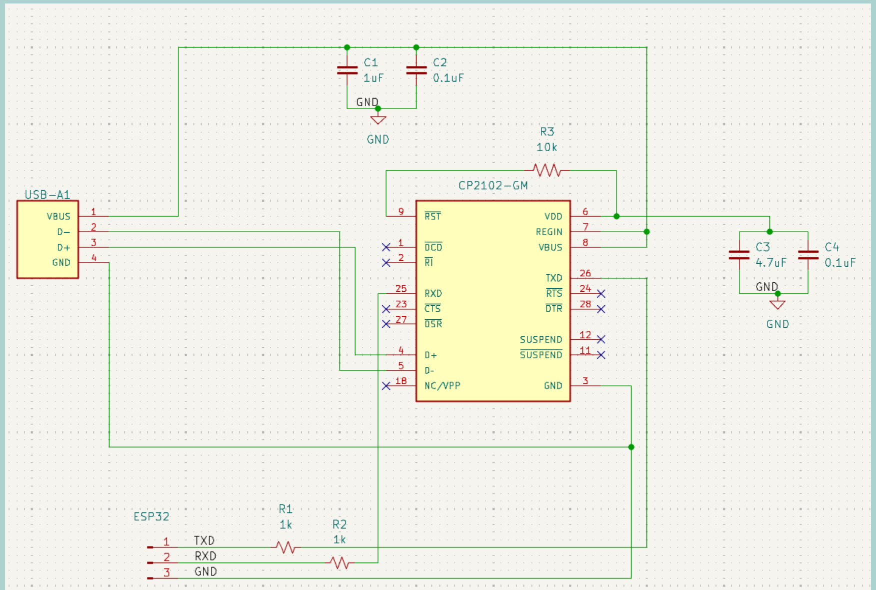






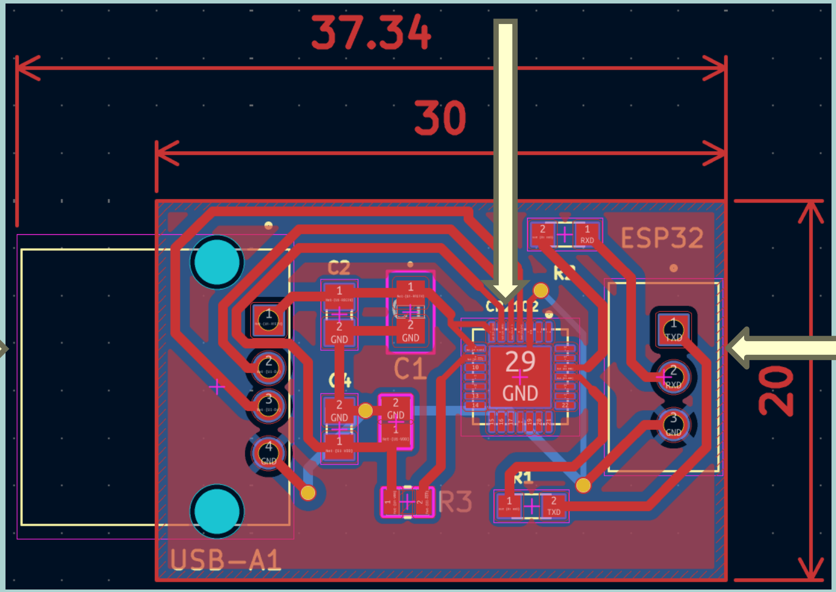
- LM386

CP2102 programming

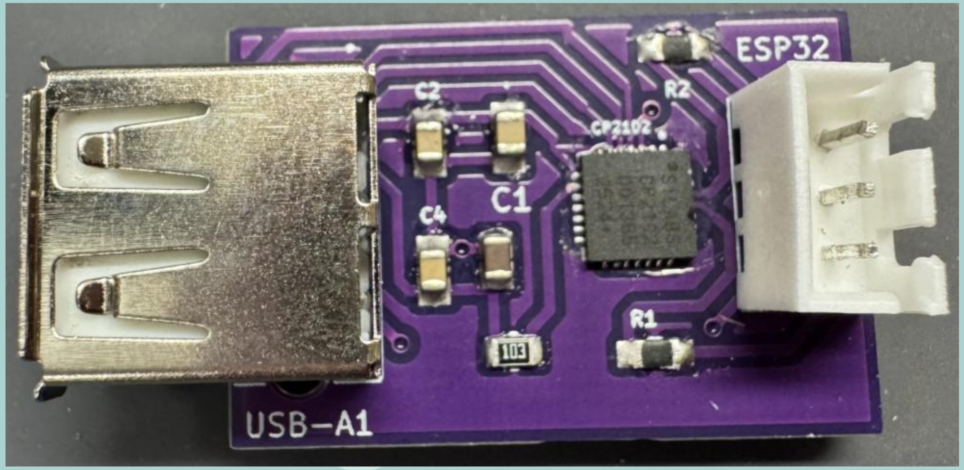


CP2102

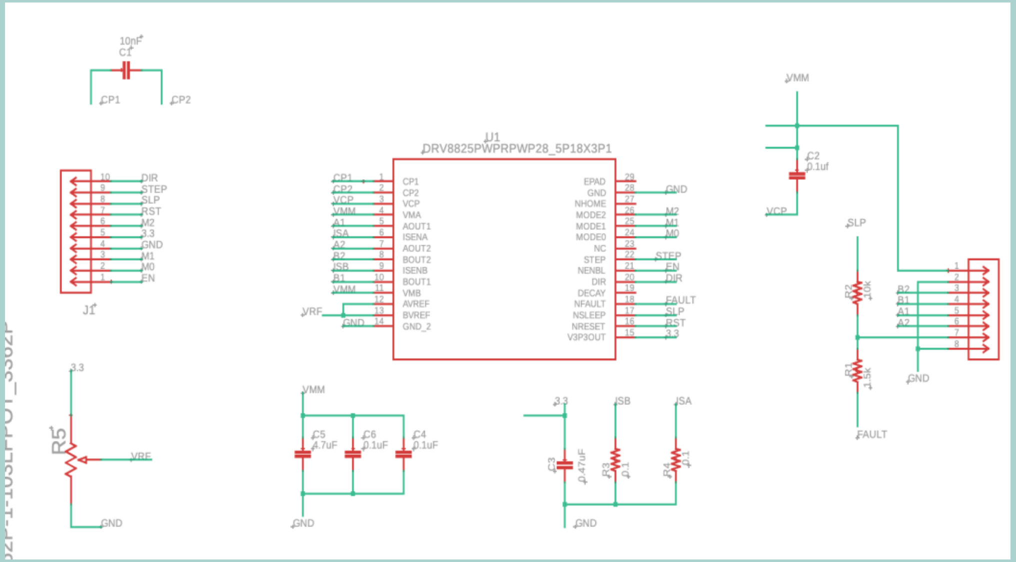
USB-A

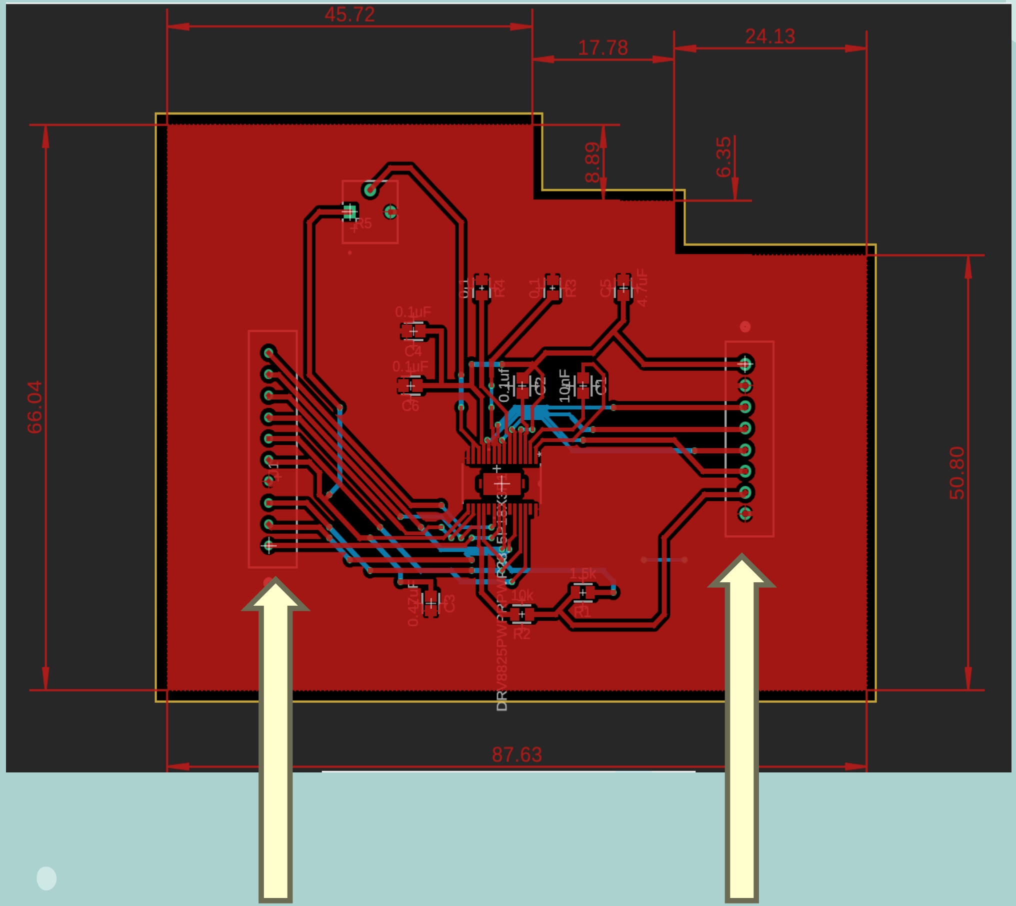


Connect to ESP32



DRV8825

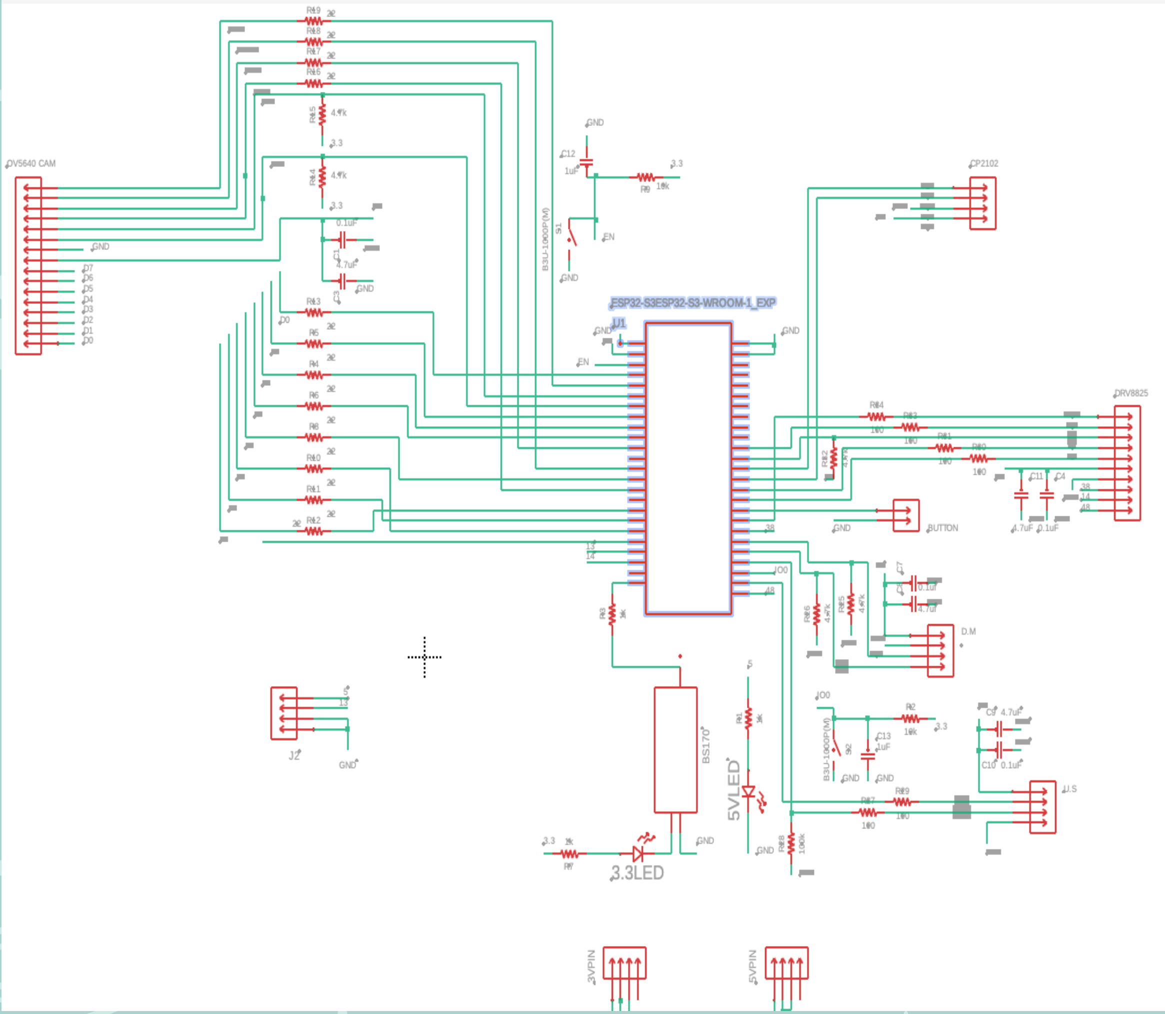




Connection to ESP32-S3 PCB

Connection to Motor

ESP32-S3 PCB



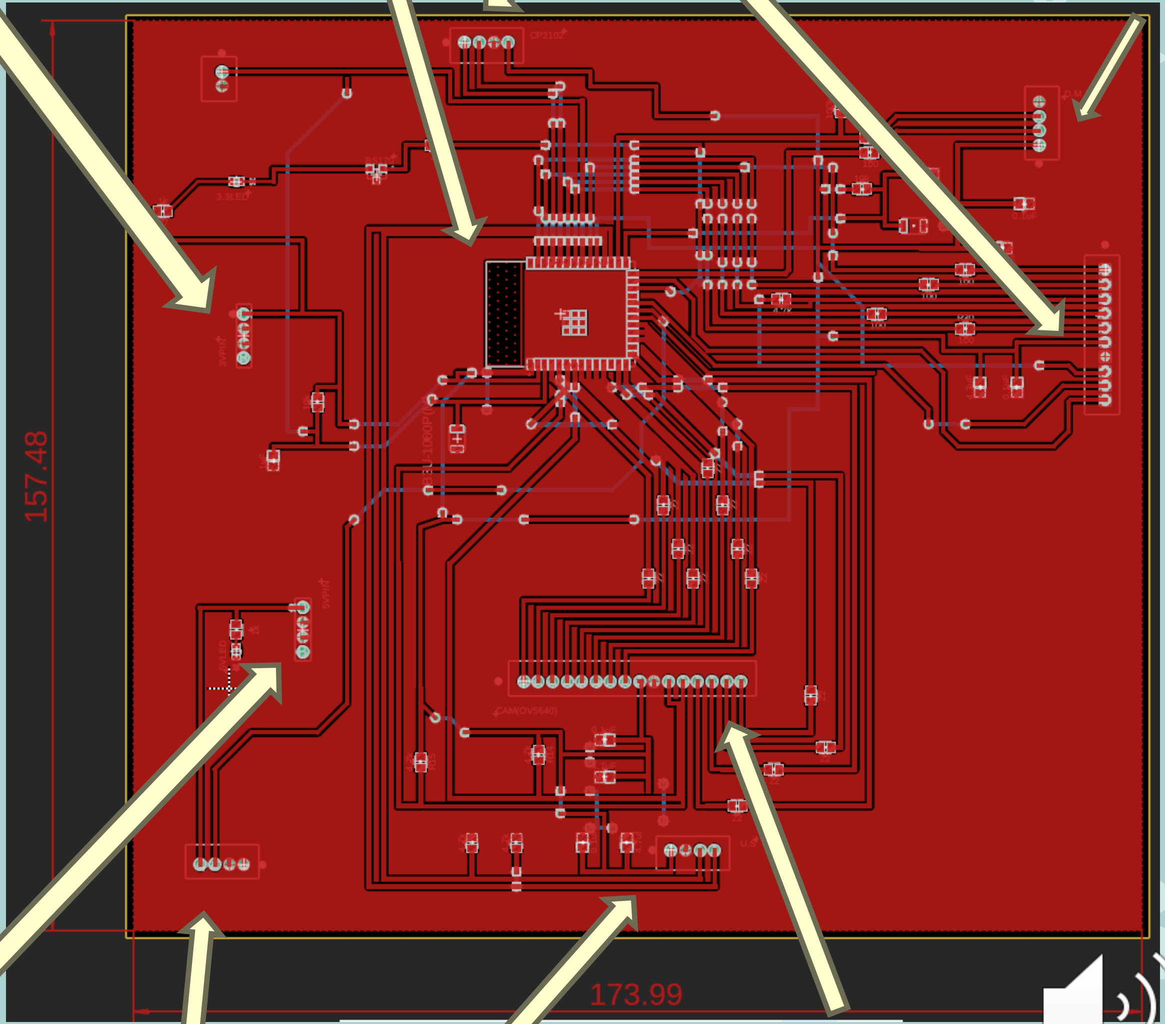
3.3V Regulator

ESP32-S3

CP2102

Motor Driver

Ultrasonic Sensor



5V Regulator

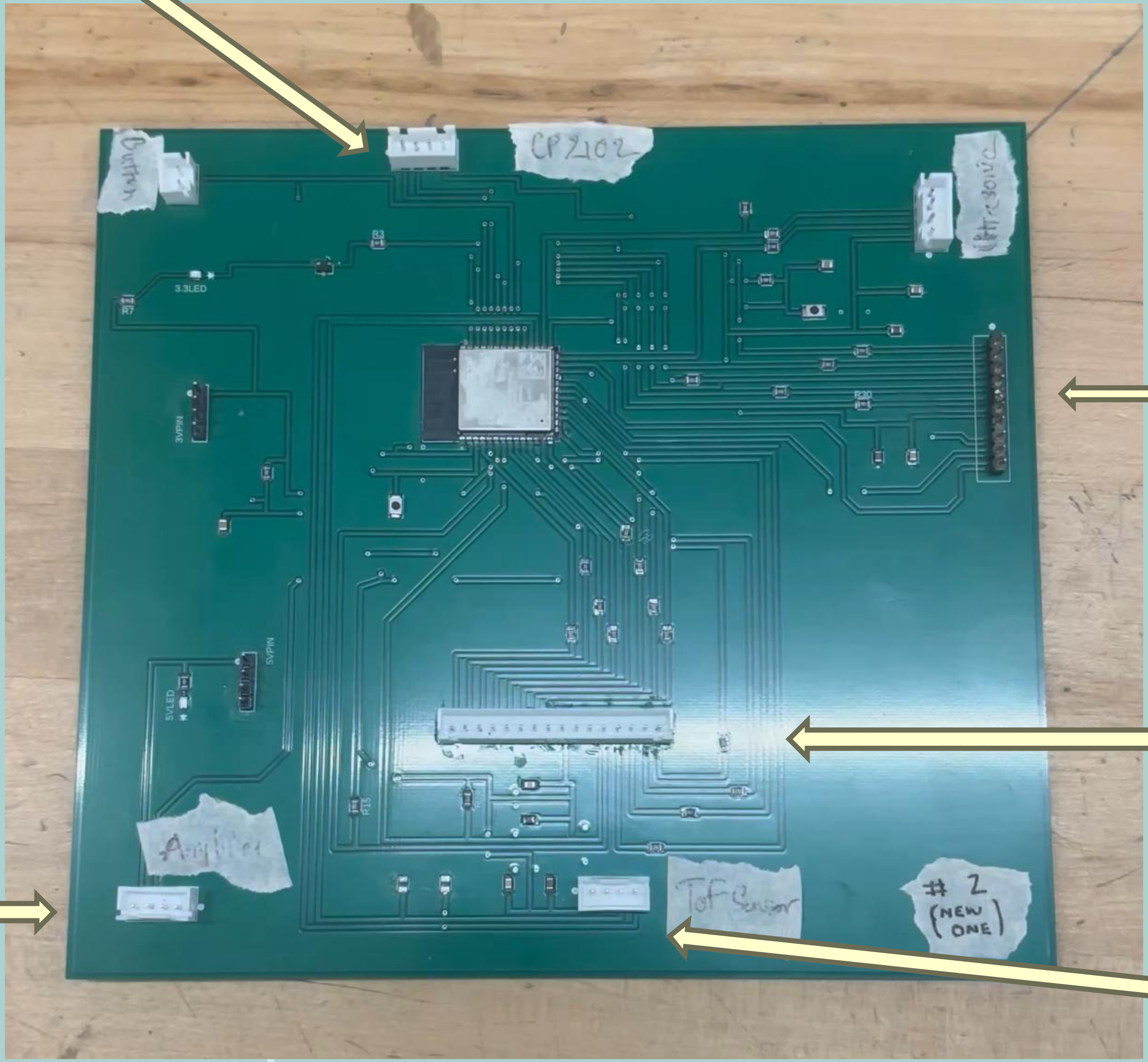
Amplifier

Depth Module

Camera

ESP32-S3 PCB

CP2102
Programmer



Motor Driver
Connecter

Camera

Amplifier
Connecter

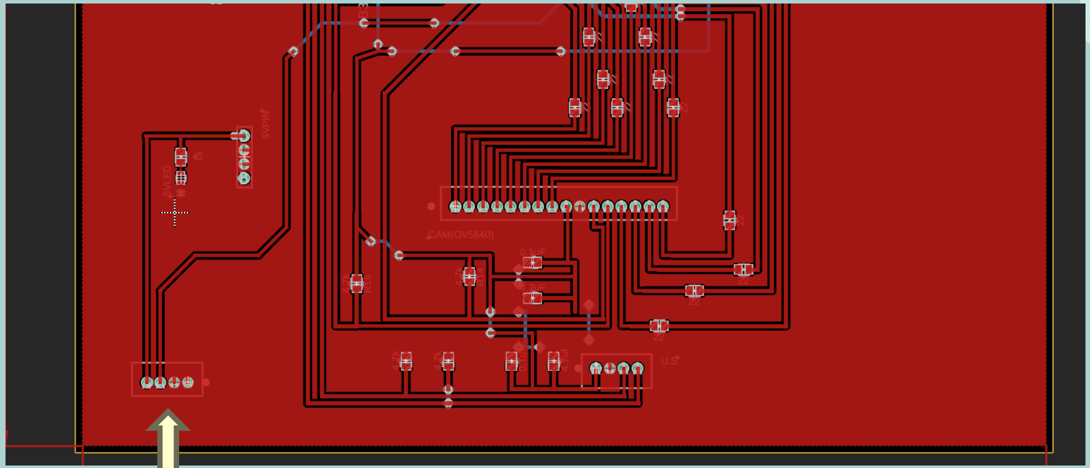
Depth
Module

Changes To The ESP32-S3 PCB

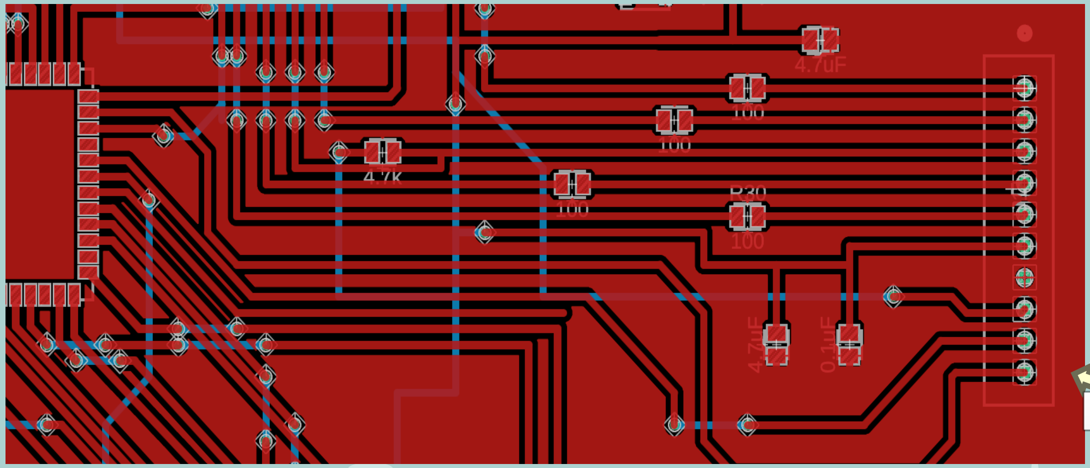
What Changed

- Amplifier Peripheral switched from the MAX98257A to the LM386 then it was removed from the ESP PCB and created as its own PCB.

- 3 extra pins were added to the motor driver header to incorporate the new DRV8825 motor driver peripheral



New Amplifier Connecter



3 New Pins

Software Comparisons

Mobile Application Stack - Frontend

- **Google-Backed & Fast-Growing Ecosystem** — Flutter's library support is expanding rapidly with strong official backing, ensuring long-term reliability and frequent updates
- **First-Class Firebase Integration** — Flutter offers the best-in-class Firebase and REST API SDK support, making backend connectivity seamless out of the box
- **Single Language, Full Control** — Dart and Flutter's widget-based paradigm give us a consistent, unified development experience across all platforms
- **Purpose-Built for Mobile** — Unlike Ionic (optimized for web devs) or React Native (JS bridge overhead), Flutter compiles natively for true performance

Category	React Native	Ionic	Flutter
Language	JavaScript / TypeScript (React)	HTML, CSS, JavaScript / TypeScript (Angular, React, or Vue)	Dart
Learning Curve	Easy if familiar with React/JS	Easiest for web developers (HTML/CSS/JS skills reusable)	Moderate — requires learning Dart and widget-based UI paradigm
Ecosystem & Libraries	Mature, vast NPM ecosystem, backed by Meta	Large plugin base, strong web developer community	Growing rapidly, strong official support from Google
Integration with Backend (Firebase, REST APIs)	Excellent — native Firebase SDKs and REST API support	Good — uses JavaScript SDKs; works well with Firebase	Excellent — first-class Firebase integration and SDKs

Mobile Application Stack - Backend

- **Fast & Beginner-Friendly Setup**: Firebase provides native Flutter SDK support and requires no server management, allowing us to deploy quickly without extensive backend experience.
- **Real-Time Cloud Synchronization**: Firestore and the Realtime Database enable instant data updates between devices, ensuring a responsive and seamless mobile app experience.
- **Integrated Storage & Cost Efficiency**: Firebase Storage securely handles images and video snapshots, integrates with Cloud Functions and analytics, and offers a generous free tier with scalable pay-as-you-go pricing.

Category	Firebase	Supabase	AWS Lightsail
Ease of Setup	Easiest setup; no servers to manage; seamless with Flutter and mobile SDKs.	Simple setup with SQL-like structure; integrates quickly with Flutter via REST or SDKs.	Requires manual API and server setup; more configuration needed for databases and hosting.
Real-Time Communication	Built-in real-time sync with Firestore and Realtime DB.	Supports real-time updates using PostgreSQL’s replication features.	Must implement custom WebSockets or MQTT for real-time data.
Database Model	NoSQL (Firestore) and JSON structure; ideal for live sync.	SQL (PostgreSQL) with relational querying and constraints.	Choose any (MySQL/PostgreSQL/MongoDB) — manual setup required.
Cost	Free tier + pay-as-you-go; cost increases with high read/write frequency.	Generous free tier; predictable pricing.	Fixed monthly per instance; predictable but may cost more idle time.

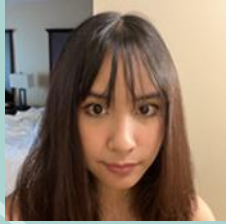
Embedded Software

- **Deterministic Real-Time Control:** C provides precise timing and predictable execution, ensuring near immediate responses when the dog is detected sitting and the motor, time of flight sensor, or speaker must activate.

- **Low Memory & High Efficiency:** With the ESP32-S3's limited RAM and processing power, C's low overhead and direct memory control make it ideal for efficient embedded execution.

- **Direct Hardware Access:** C allows direct interaction with registers, drivers, and peripherals, enabling accurate control of the motor, camera, LEDs, speaker, and Wi-Fi module.

Category	C/C++ with FreeRTOS
Use Case	Low-level embedded control for MCU. Run components concurrently.
Advantages	Direct access to hardware registers and peripherals. Runs efficiently with minimal memory. High performance and timing precision. Real-time scheduling ensures deterministic behavior. Enables parallel task management.
Disadvantages	Manual memory management. Limited abstraction. Longer development cycles. Requires careful task priority tuning. More complex.

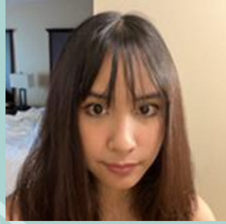


Cloud Server

Feature	Google Cloud Run
Setup	Easy
Server Management	None
Auto-Scaling	Yes
Best For	REST APIs
CPU Control	1-8 vCPU (selectable)

- **No sever management:** Google Cloud Run does not require any server management, so no Linux maintenance, firewall configuration, or patching.
- **Built for HTTPS interface APIs:** Allowing for an easy integration with Firebase and request routing is handled automatically.
- **Auto-scaling:** It scales automatically for any future upgrades or multiple feeders, and if camera or audio usage becomes frequent.
- **Simple deployment:** Build a Docker container or connect a Github repository, deploy, and get HTTPS URL.
- **Other Methods:** Require server maintenance and setup while providing less flexibility.

Detection Model



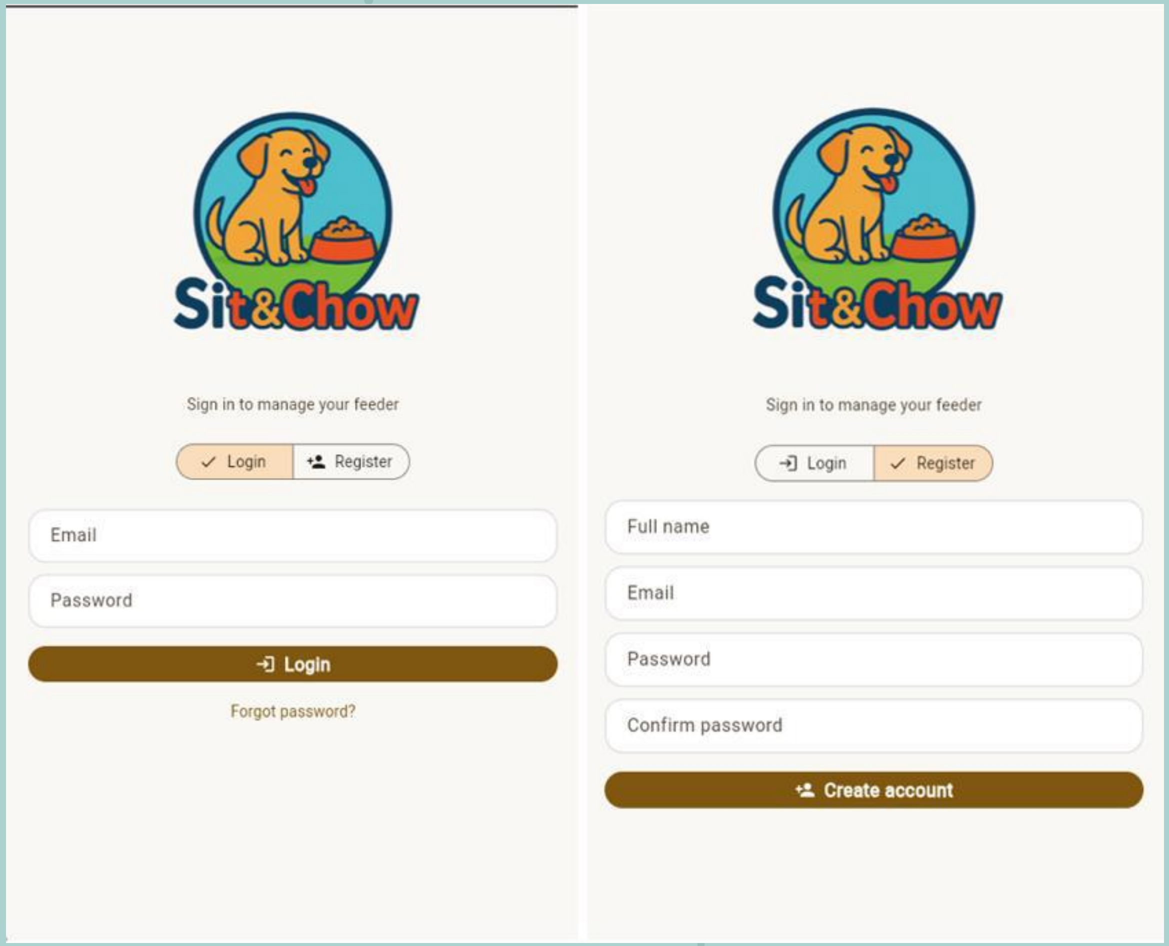
Feature	YOLO (You Only Look Once)
Real-Time Speed	Enables continuous live detection from camera stream
High Accuracy	Already trained to detected general objects, like dogs
Scalability	Multiple model sizes and variants for device efficiency

- Fast, real-time detection.
- Has lightweight models, so it can run well on CPU.
- Pretrained on object classes, including dogs.
- Other models would be too slow, less accurate, and reliable for real-time streaming

Software Design

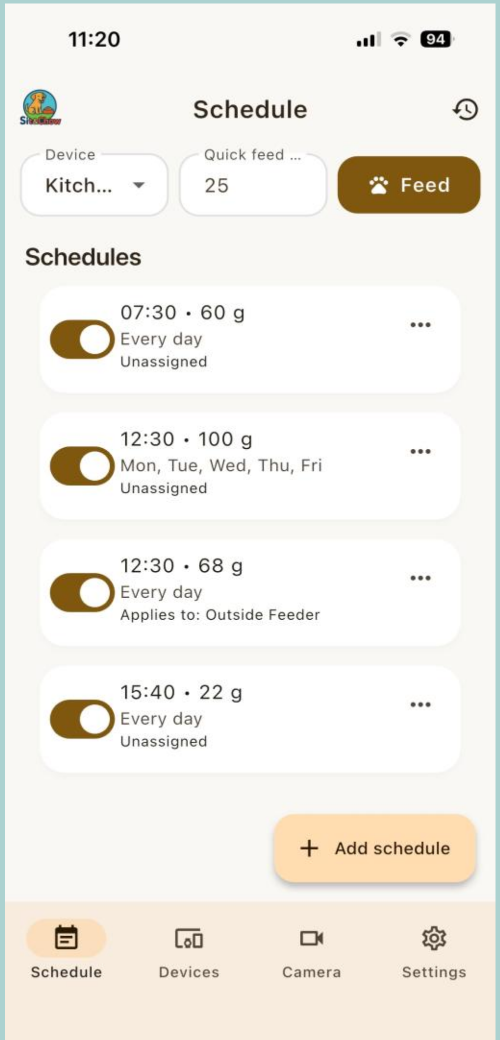
Mobile Application

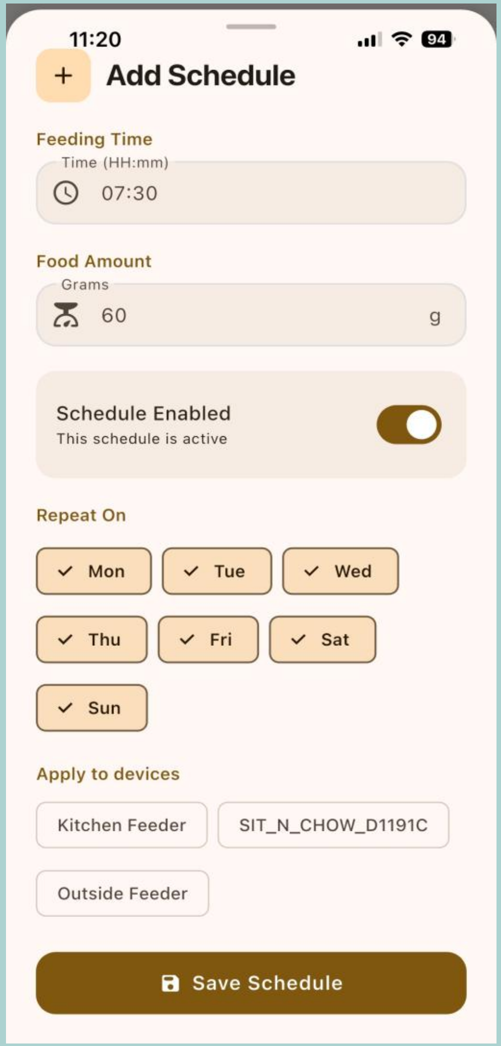
Mobile Application – Login and Register



- The first page that the user sees is the login and register page
- When they register, the information they enter is sent to the Realtime Database in Firebase to authenticate the user in the login process
- Note there is also a forgot password function on the login page should the user need to use it
- Once they register or login, they are sent to the landing page

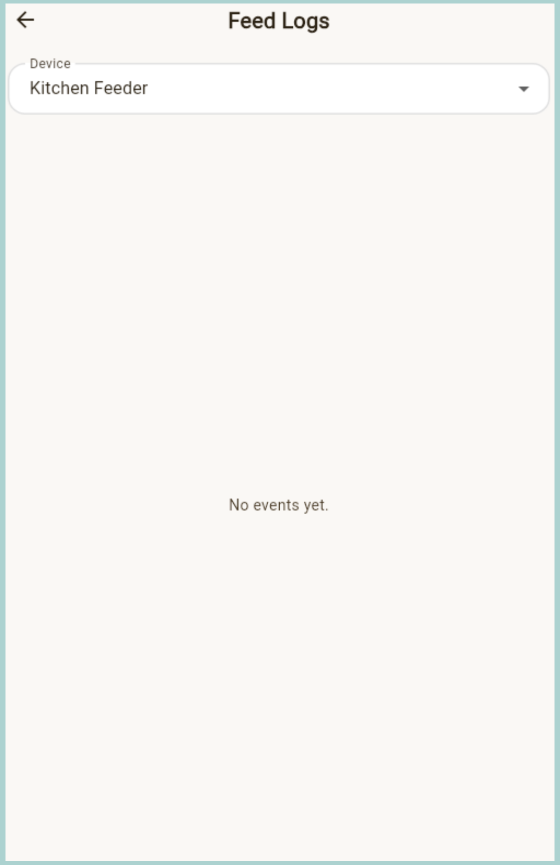
Mobile Application – Schedules





- The schedules page displays all of the schedules the user has created
- A quick feed option is available to immediately activate the feeder, bypassing the detection model should the user need to do so
- The user can create a new schedule, assigning different days, times, food amount, and devices the schedule is assigned to
- The user can freely edit or delete the schedules created at any time

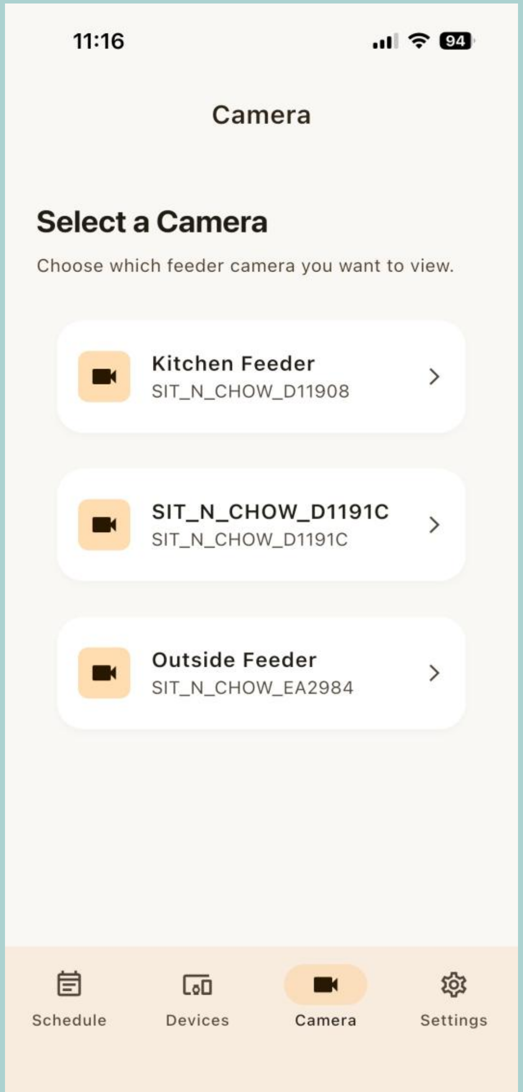
Mobile Application – Feed Logs & Devices



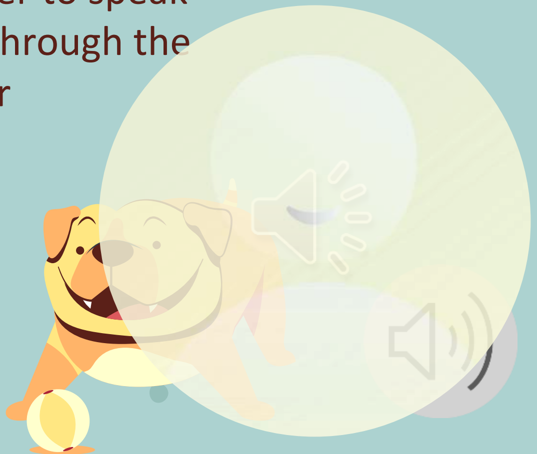
- By clicking the history icon on the top right of the schedules page, the feed logs page appears
- This page tracks the earlier feeds and the amount dispensed

- The device page displays all of the currently linked devices
- To link a device, the user will simply click on add device and that will route them to the provisioning page where they can configure the Wi-Fi the feeder is set to
- The Device ID is generated using the ESP's MAC ID and will be used during device provisioning
- The user can name the device once connected and can then freely edit or delete said device

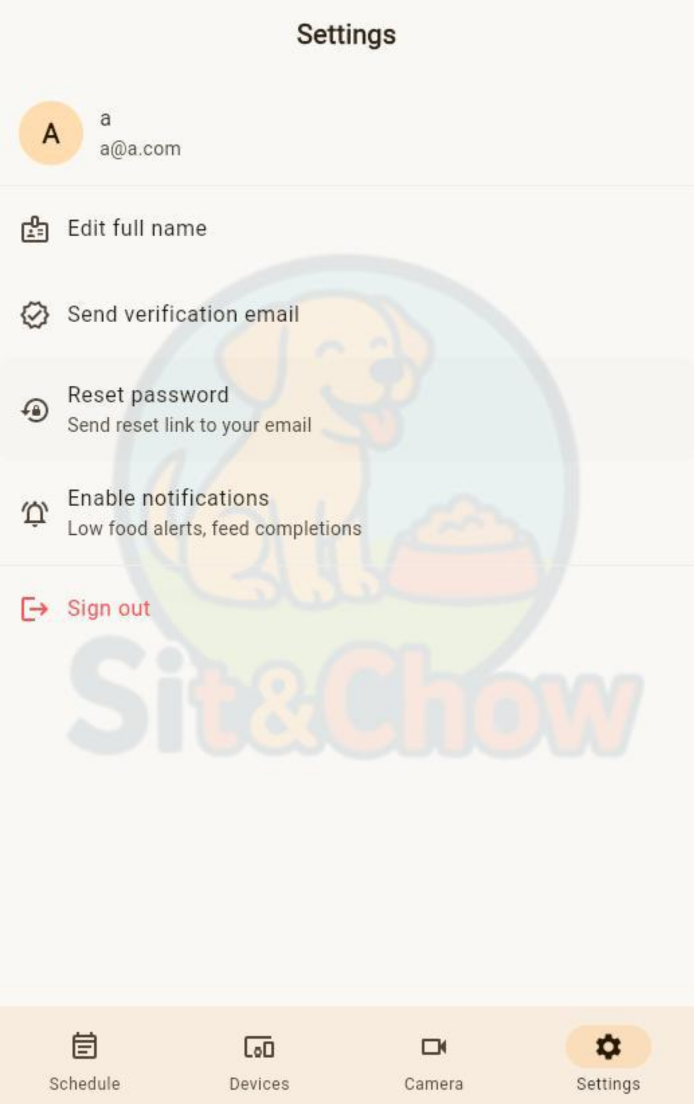
Mobile App – Live Camera Feed



- The camera page will have the list of devices so that the user can select which device's camera they want to see
- They can click on the device's name so that they can look through the camera on that device
- Once the live feed page is open, the live feed will activate, and the user will be able to see through the camera
- There is an interactable microphone symbol that allows the user to speak through their phone out through the speaker on the dog feeder



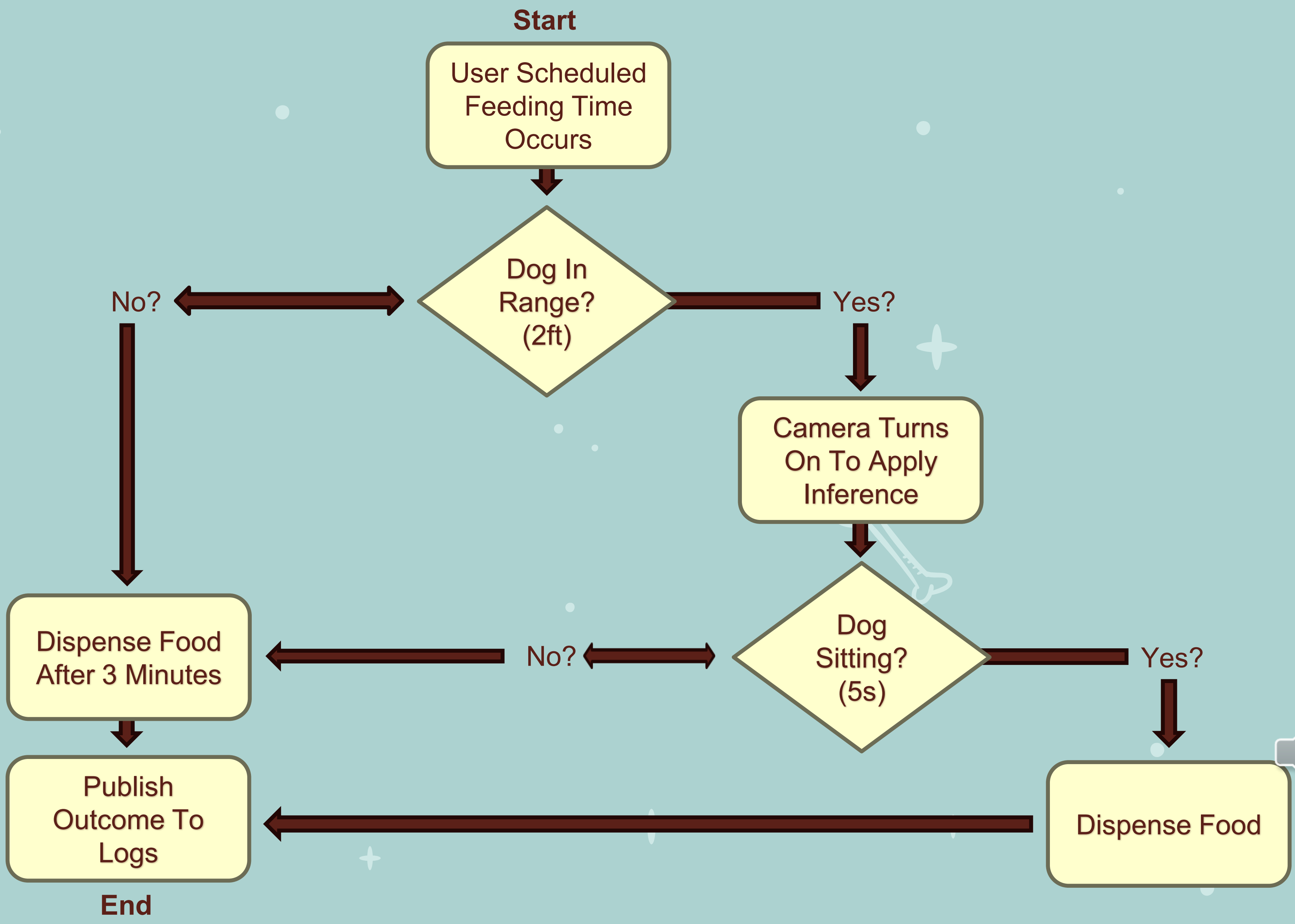
Mobile Application – Settings

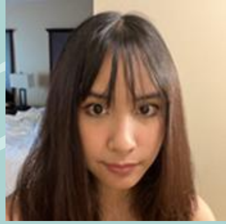


- On the settings page, the user will be able to change their full name
- They will be able to verify their email address
- A reset password function is available should the user wish to change their password
- They are also able to enable their notifications on this page if they turned it off before
- Lastly, they can sign out of their account if they wish to do so, sending them back to the login and register page

Main Feeding Workflow

Scheduled Feed Workflow

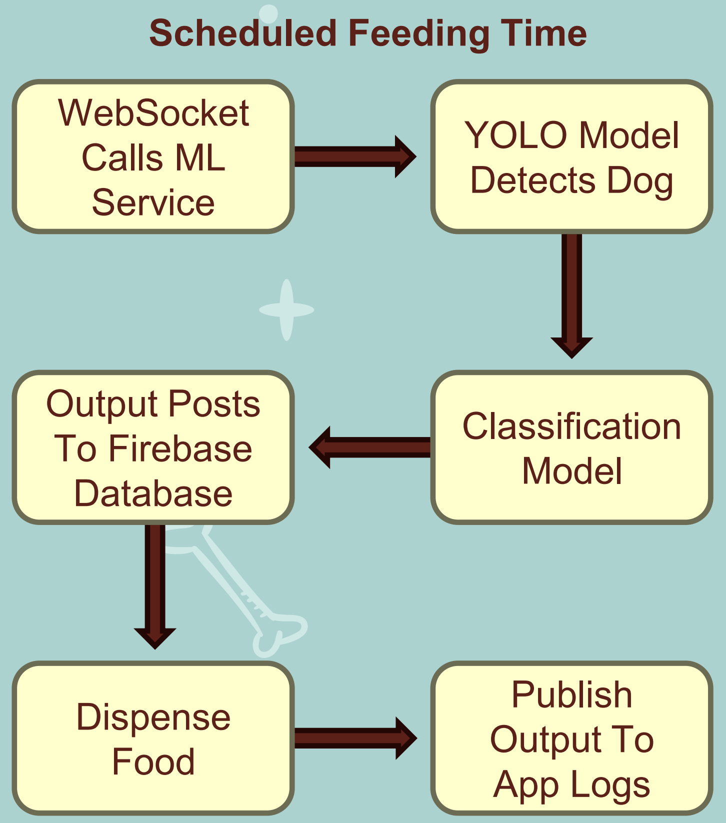


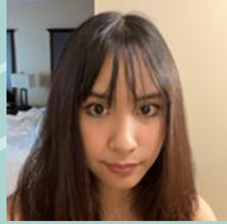


Google Cloud Run Services

2 services are deployed:

- WebSocket:
  Streams camera frames to the mobile app and calls the machine learning prediction for each frame.
  Streams microphone audio to speaker.
- Machine Learning:
  Holds the detection and classification pipeline.





Posture Detection and Classification Pipeline

The detection and classification pipeline operates in 5 main stages:

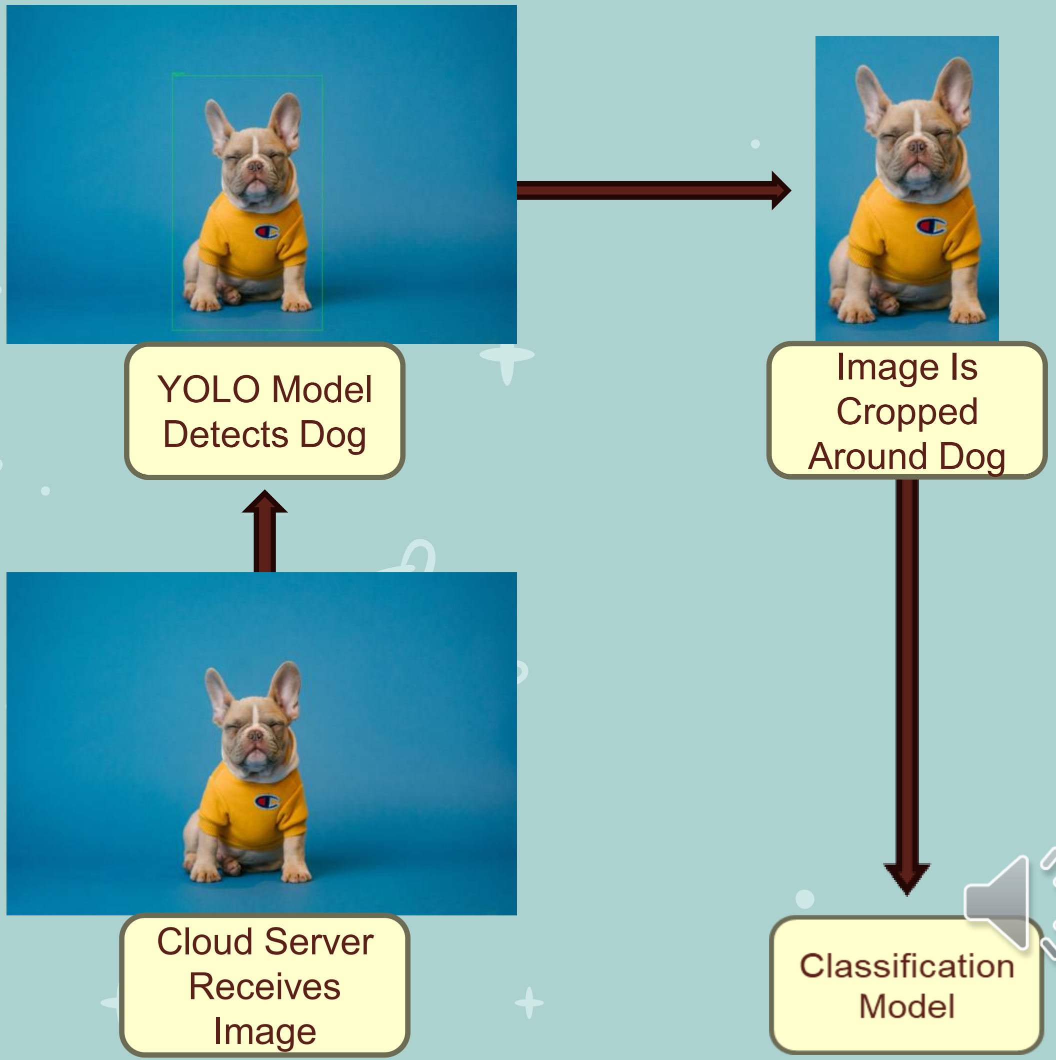
- Getting the image
- Detecting the dog
- Displaying and cropping the bounding box
- Classifying the posture
- Executing the event

YOLO (You Only Look Once) is a computer vision model that is pretrained on the COCO dataset. Dogs are one of the objects it has learned to detect.

- A box will be drawn around the dog, and the image will be cropped around the box.

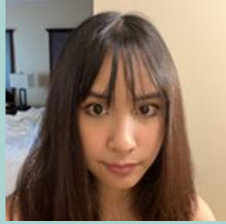
The new cropped image will be passed through a custom binary classification model to determine if the dog is “sitting” or “not sitting.”

- The classifier uses a ResNet-18 backbone with a single-logit output layer and trained on a combination of Stanford Dogs images and images captured the system’s camera.

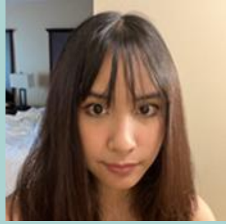


Prototyping and Testing

Hardware



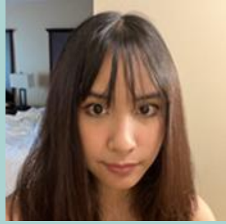
- **Camera Output**: 1st main PCB iteration had hardware connection issues where the output was very distorted, causing problems for the ML to run properly.
- **GPIO Pins**: Two GPIO pins connected to the time-of-flight sensor and one GPIO pin connected to the ultrasonic sensor were accidentally wired to pins held by PSRAM. If we wanted to use PSRAM, we would have to free those pins. We solved this by moving three pins meant to set the microstepping on the motor driver and rerouting the ToF sensor and ultrasonic sensor to those newly freed pins.
- **CP2102 Programmer**: Since the chip is too small to solder, we only have 2 of them that work properly.
- **Voltage Regulators**: work well and don’t have problem
- **Motor**: The connectors are smaller than we expected, we need to solder wires to the pins on the motor driver board to the main board. We need to be careful not break the wire connections.



Google Cloud Run

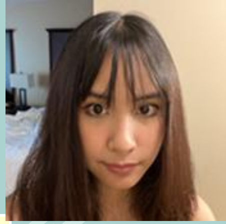
- Successfully integrated Firebase, the mobile app, and machine learning between 2 services, 1 for machine learning and the other streaming the camera and microphone audio.
- Problems:
  - Microphone audio output: Mismatched endpoints had to be corrected. 1 endpoint to store the microphone input, and 1 to output the audio to the speaker.
  - Deployment: WebSocket Service failed to deploy without the proper environment variables, such as missing dependencies, Firebase credentials, and API key.
  - Database output: ensuring the final output is written to the correct location and not overwriting pre-existing mobile app functionalities.

Machine Learning



- Pipeline Performance
  - ML pipeline performed well before connecting to the system’s camera. However, there were some struggles classifying the posture when the dog is directly facing the camera, so more training images of that view needed to be added. Standardizing the image size and adding padding and clamping to the cropped images ensures all features within the detection bounding box were seen for the classifier.
  - YOLO successfully detected dogs on the system’s camera. Detection accuracy was lower using the system’s camera, but not enough to hinder overall performance.
  - Classifier performance struggled with the system’s cropped images for sitting poses. Therefore, “sitting” and “not sitting” images had to be taken with the system’s camera to retrain the classifier.
  - The final ML output successfully stored in the database to be accessed by the mobile app and ESP32-S3.

Administrative Content



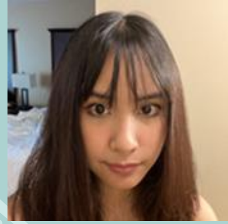
Contributions

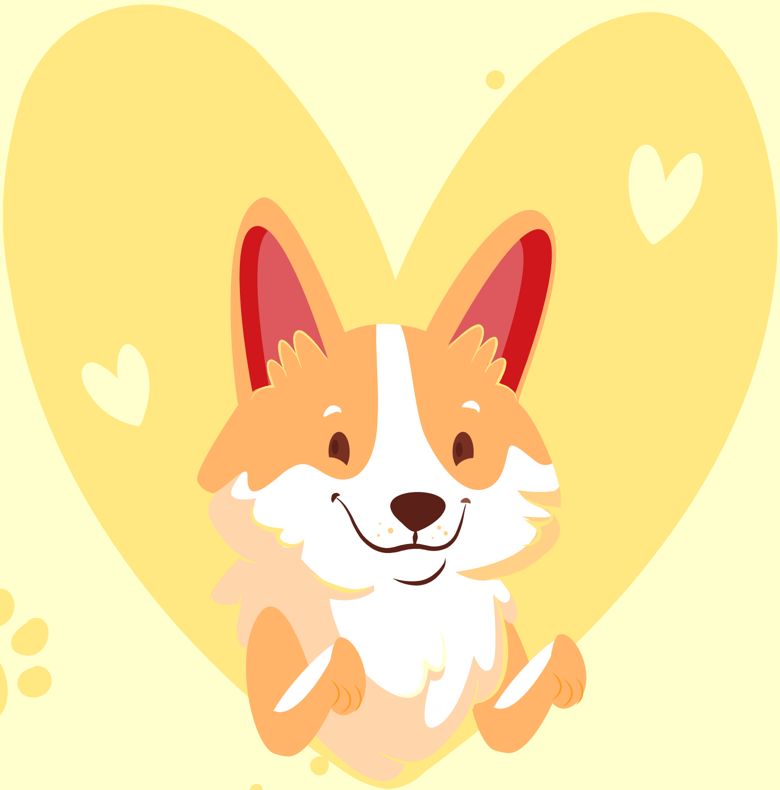
Senior Design 1	
Andres	Project Description, MCU Selection, Mobile Communications, MCU and Mobile Application Software Design, Case Study, Software Testing
Andrew	Hardware Selection (Camera), Standard Constraints, Hardware Design, PCB
Maria	Introduction, Project Description, Hardware (Depth Module), Detection and Classification Model, Software Constraints, Software Testing, BOM
Natalie	Hardware Selection (Motor, Motor Driver, Speaker, Amplifier, LEDs, Button, Sensing Storage Capacity, Power Management), Case Study, Hardware Design, PCBs, Hardware Testing, BOM, Milestone
All	HOQ, Specification Table

Senior Design 2	
Hardware	
Andrew	Main PCB, Motor Driver
Natalie	Power Supply System, Amplifier, Programmer, 3D Model for Case
Software	
Andres	Embedded Firmware, Mobile Application, WebSocket Service
Maria	Detection and Classification Pipeline, WebSocket Service, Machine Learning Service
All	Full System Integration

Budgeting

Component	Quantity	Price Total
Stepper Motor	1	3.33
Motor Driver	1	4.59
Power Adapter	1	9.99
Barrel jack	1	0.65
Camera	1	13.76
Voltage Regulators	2	18.64
Amplifier	1	5.95
Speaker	1	1.95
Programmer (CP2102)	1	8.88
USB Port	1	0.50
Ultrasonic Sensor	1	3.95
Depth Module	1	13.00
ESP32-S3	1	10.67
PCBs	5	252.83
PLA	6	125.94
Total Cost		**~474.63**





THANK YOU!

Group 25 wish to thank for the guidance throughout this senior design project:

- Mentor: Dr. Saleem Sahawneh
- Review committee:
  - Dr. Tong Wu
  - Dr. Shahana Ibrahim
  - Dr. Truong Nghiem
- Lab professor: Dr. Arthur Weeks