

ECE 445
SENIOR DESIGN LABORATORY
FINAL REPORT

Dodgebot

Team #41

LOIGEN SODIAN
(sodian.21@intl.zju.edu.cn)
JADEN PETERSON WEN
(peterson.21@intl.zju.edu.cn)
QINGYAN LI
(qingyan5@illinois.edu)
PUTU EVAITA JNANI
(putu.21@intl.zju.edu.cn)
ISAAC KOO HERN EN
(ikoo2@illinois.edu)

TA: Zhefeng Guo

May 16, 2025

Abstract

This report details the development of an automated dodgeball machine capable of launching tennis-sized projectiles at targets up to 5 meters away. The system integrates dual 48 V motors (6000 rpm, 0.15 Nm torque) with thermal imaging technology for real-time target detection and tracking. The machine processes thermal data to identify target centroids, converting this information into PWM signals for the servos. A novel spiral magazine system provides efficient ball storage while being aesthetically pleasant. Overall, this system demonstrates significant potential as a commercially viable product.

Contents

1	Introduction	1
1.1	Problem and Solution Overview	1
1.1.1	Visual Aid	1
1.2	High-level Requirements List	2
1.3	Report Outline	2
2	High-level Design	3
2.1	Block Diagram of <i>Dodgebot</i>	3
2.2	Operation Control Module	4
2.3	Machine Vision Module	4
2.4	Turret Module	5
2.5	Firing System Module	5
2.6	Power Module	6
3	Operation Control Module	6
3.1	Design Verification	6
3.2	Panic-pause Feature Description	6
3.3	On-the-Air (OTA) Flashing Capability	7
3.4	Conclusion	7
4	Machine Vision Module	8
4.1	Object Detection Algorithm Using Trinary Mapping and Weight Classifica- tion	8
4.1.1	Image Polling Through Websocket	9
4.1.2	Centroid Detection Algorithm	9
4.2	Motor and Servo Control	10
4.3	Design Verification	12
4.4	Conclusion	12
5	Firing System Module	13
5.1	Controlling the Firing Motors via PWM	13
5.2	Structural Design of the Gun and Magazine System	14
5.2.1	Gun Platform	14
5.2.2	Gun Module	14
5.2.3	Magazine	15
5.3	Design Verification	15
5.4	Conclusion	16
6	Power Module	16
6.1	Power Distribution Using Step-Down Converters	17
6.2	PCB Design for ESP-12F and AMS1117	18
6.3	Design Verification	19
6.4	Conclusion	19

7	Cost Estimation	20
7.1	Expenses on Parts	20
7.2	Labor Reimbursements	20
8	Conclusion	20
	References	22
	Appendix A Machine Vision R&V Table	23
	Appendix B Firing System R&V Table	25
	Appendix C Machine Vision R&V Table	27
	Appendix D Code Listings for Operation Module	28
	Appendix E Detailed Expenses	30
	Appendix F Power Schematic and BOM	32
	Appendix G Base and Turret Schematic Design	35

1 Introduction

In this chapter, we discuss the underlying motivation that leads to the final decision in creating a dodgeball launcher and the high-level requirements that we set as a measure of success when compared with the final design. Additionally, Section 1.3 explains the structure of this report.

1.1 Problem and Solution Overview

The traditional sport of dodgeball, while popular, faces significant limitations in terms of accessibility and safety. Since sport requires two teams to play, it takes at least two people to operate, and the high-speed throws pose a risk of injury, especially to vulnerable areas. Recently, there have also been new technological developments exploring new forms of dodgeball play, such as augmented reality (AR) [1] and motion tracking devices [2] that attempt to improve the sport but also present new obstacles. Augmented reality relies on controlled environments and wearable sensors, limiting spontaneity, while motion tracking systems perform poorly in terms of latency and accuracy under variable lighting conditions [3]. There are still a number of issues that need to be addressed in developing a robotic device platform that can mimic the mechanics of a dodgeball game while ensuring accuracy and safety, including instant target detection, turret mobility, and more.

In response, we propose an improvement program that aims to address these challenges by integrating thermal cameras, mechanical turrets, and an autonomous robotic system with AI modular design. For immediate target detection, unlike AR or motion tracking, *Dodgebot* employs a thermal camera compatible with Espressif's ESP-12F [4] for real-time body detection without relying on wearable sensors. For the motorized turret, the Arduino-driven 360° turret module allows for full rotation while avoiding wire entanglement to ensure unrestricted movement. The firing mechanism utilizes dual counter-rotating motors to propel the projectile, reducing the risk of injury by controlling speed. The solution is designed to be powered by a 48 V max mobile power supply, ensuring portability and scalability. This approach improves previous systems by combining accessibility, environmental adaptability, mechanical precision and safety to provide a novel solution.

1.1.1 Visual Aid

The robot will function as shown in Fig. 1. In an ambient environment, there may exist small deviations in temperature between each pixel of the thermal camera. Hence, the robot (controlled through the ESP-12F MCU) will deem the target non-existent in this context. If so, the robot will rotate to a new firing position, to which if an object of appreciable size (which will skew the temperature measurement) is detected, the MCU will first calculate the center of the "blob", and then aligns the turret to the object's center. It will then immediately fire once the gun is aligned to be about the center of the object (not necessarily need to be precisely in the middle).

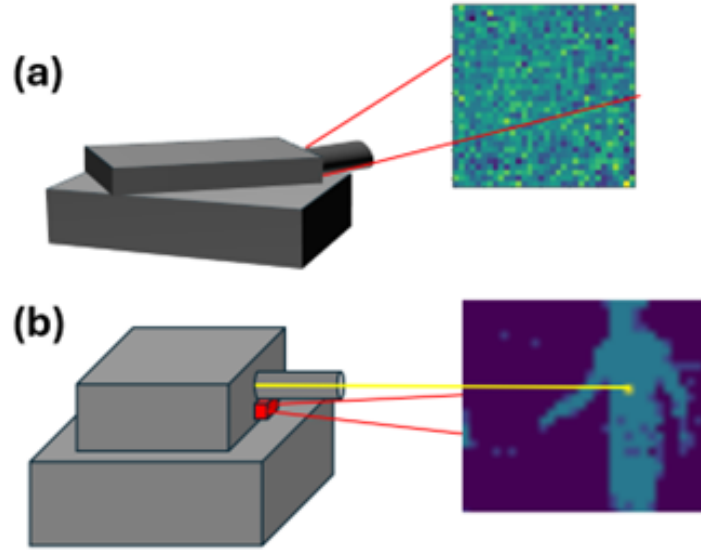


Figure 1: The operation mechanism of the robot. During (a), it scans the environment ahead using a thermal camera attached to the underside of the gun barrel (red box, with red lines as its view area). If a target is not seen, it will rotate to a new direction. If a target is detected, the turret aligns with the object's center of mass and fires immediately.

1.2 High-level Requirements List

The following are the requirements to which will be evaluated against the finished design:

- The turret module should be able to rotate fully to 360 degrees once within 20 seconds. Hence, wiring must be done inside the robot to ensure that it will not be entangled when the turret rotates.
- The firing mechanism should be able to fire without misfiring $\geq 95\%$ of the time.
- The robot must achieve $\geq 80\%$ firing accuracy when targeting moving humans within a 5-meter range under standard (flat terrain, good visibility) conditions.
- The machine vision classification should achieve accuracy of 80%.
- The whole design should be modular, except power delivery, which may require external outlets. This means no connection is allowed between the MCU and a computer.

1.3 Report Outline

Section 2 discusses the high-level description of each module. Section 3, Section 4, Section 5, and Section 6 discusses each components: operational control, machine vision, firing system, and power distribution accordingly. Section 7 details the total expenses of this project, and finally Section 8 concludes this paper.

This chapter primarily covers the high-level design of *dodgebot*, along with a discussion detailing the purpose of each components.

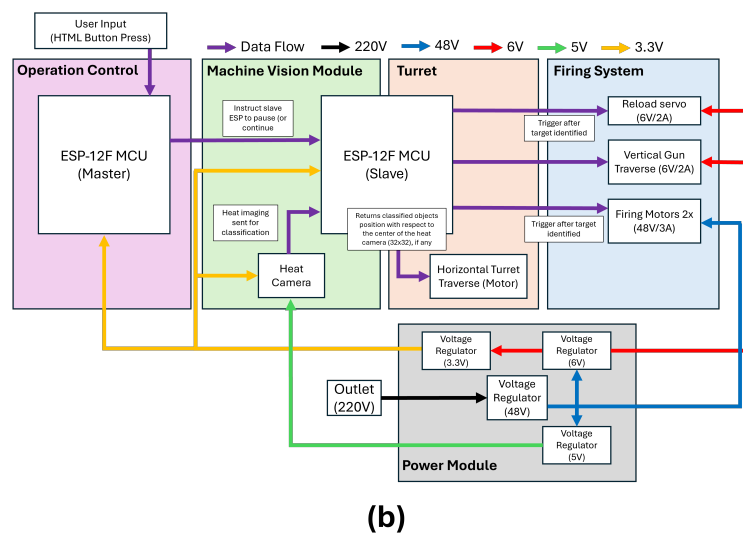


Figure 2: (a) shows the 3D schematic of the robot. Red represents power delivery devices, green represents the operational and machine vision aspects of the robot, and blue represents the firing system. Casings/structures are colored gray. (1) is the 48 V to 6 V DC-DC converter, (2) is the 48 V to 5 V DC-DC converter (to power the thermal camera), (3) is the gun mount, (4) is the vertical traverse servo, (5) is the horizontal traverse servo, (6) are the firing motors, (7) is the reload servo, and (8) is the fired ball. The "PCB" block consists of the ESP-12F module along with the 6 V to 3.3 V DC-DC converter. Colored traces follow the legend shown in (b). (b) outlines the schematic for all the modules required to achieve the design.

The 3D illustration and the block diagram of each components are shown in Fig. 2a and Fig. 2b respectively. The robot fires a tennis ball (diameter 63.5 mm) using two, 6000 rpm, 48 V DC motors. To fire the ball, the heat camera first supplies data through Wi-Fi to the PCB component (which is comprised of two ESP-12F MCUs). The slave ESP then

determines whether an object is present, and if present, it will trigger appropriate signals to initiate firing. Depending on the devices' function in Fig. 2a, we can separate them into five different modules i.e., *operation control*, *machine vision module*, *turret*, *firing system*, and *power module*, shown in Fig. 2b.

2.2 Operation Control Module

This module determines whether the slave ESP should initiate its routine target detection and control of the firing system. This module creates an access point (AP), to which the user may connect to if they wish to control the state of the robot. Through an HTML interface provided by this module, the user can either pause or continue the robot by pressing a button using their mobile device. Additionally, a status message detailing the robot current status (i.e., active or inactive) is shown in the webpage to improve clarity. If the robot is actively engaging targets, pressing the button deactivates the robot. Likewise, an inactive robot will be reactivated when the button is pressed again.

2.3 Machine Vision Module

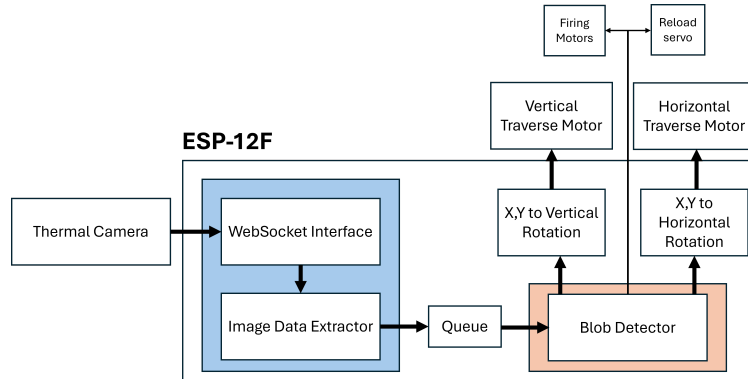


Figure 3: Block diagram of the machine vision module. Blue represents the data collection process, and orange represents the data processing process.

This module consists of a thermal camera and an ESP-12F to perform image classification, with the block diagram shown in Fig. 3. The camera is mounted on the gun barrel; hence it is aimed exactly where the gun is aiming at. On a specific viewpoint, the ESP module will first identify objects with heat signatures higher than the ambient environment. We implement this by calculating the difference in temperature between one pixel and its neighboring pixel and determining whether this is a possible target if the difference is substantial. If a target is detected, the slave ESP will send instructions to traverse the gun both horizontally and vertically towards the target, reload and fire the gun. To determine the amount of rotation needed, the distance between the centroid of the target and the camera's pixel array center is calculated and is then converted into degrees, a more readable unit for the servos. To determine the angle offset, we obtain the angle needed to offset an image by one pixel, and extrapolate this for various pixel offsets. If there are no targets, the ESP will rotate the turret to a new viewing point for image classification.

2.4 Turret Module

The turret module houses the firing system. To rotate the turret, the ESP MCU (from machine vision module, Section 2.3) will send instructions to both the horizontal and vertical traverse motors to rotate the turret and elevate the gun so that the target is centered. The elevation to pixel occupation relation is tabulated manually, similar to Section 2.3. If no target is detected, the ESP will send instructions to rotate the turret to another angle for image scanning by the machine vision module. To fire the gun, the ESP will activate the reload motor once to allow a single tennis ball into the chamber. When the ball enters the chamber, the ball will be in contact with the firing motor (which is rotating in-plane, one clockwise and another counterclockwise), which will launch the bullet outwards. The machine will continuously track and fires at the target until the target is out of the robot's view. After the robot stops firing, the ESP will rotate the turret to another viewpoint for classification, return the gun elevation to normal, and switch off the firing motor.

2.5 Firing System Module

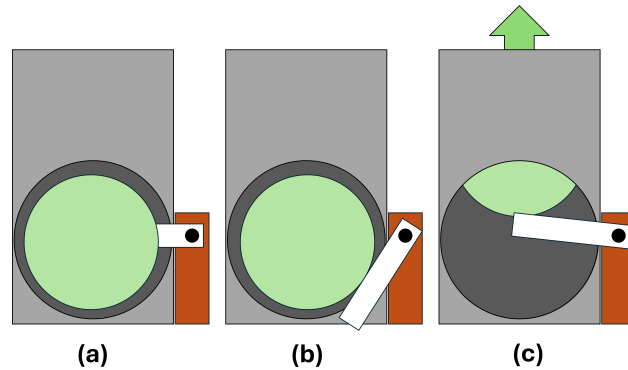


Figure 4: Reloading process of the robot. The chamber is initially empty, and the ball rests on the reload servo's spoke (a). When reloading, the spoke turn backwards to let the ball into the chamber (b). The loaded ball is launched into the firing motors in (c).

The firing system mainly consists of motors which control the ball reloading mechanism, the ball firing mechanism, and the gun elevation mechanism. The ball reload mechanism is controlled by an MG996R servo with extended spokes (reload motor), which regulates ball intake into the chamber, and ensures that only one ball is loaded into the machine at a time. Fig. 4 illustrates the ball reloading mechanism, which is comprised of three stages: initial position before reload (Fig. 4a), load the ball into the chamber (Fig. 4b), and push the ball towards the firing motors (Fig. 4c). The firing mechanism consists of two horizontal motors placed beside each other, with a gap that barely fits the bullet diameter. The two motors rotate in opposite directions, each rotating at 6000 rpm with 0.16 Nm torque to ensure that the ball does not get jammed in the chamber and is propelled at high speed. Finally, the elevation mechanism is controlled by a single servo, capable of lifting 25 kg, which is attached directly to the gun (gun barrel and the two firing motors), and the reload tube.

2.6 Power Module

This module provides power to the motors and ESP MCU. It will be powered by a wall outlet which will be distributed to all the devices on the system using voltage regulators. An AC-DC converter (i.e., S-500-48) is first used to convert AC to 48 V of DC power. Some of the power is directly used to spin the firing motors, and the rest is lowered down to a lower voltage to power the other components. We implement a “trickle down” system where we gradually step down the voltage from 48 V (DC) to either 6 V or 5 V. The reload servo along with the horizontal and vertical traverse servos are powered using 6 V, while the thermal camera is powered using 5 V USB. Additionally, we stepped down the voltage from 6 V to 3.3 V in order to power the two ESPs using the AMS1117-3.3.

3 Operation Control Module

This section details the *operation control module*, starting from the design verification in Section 3.1, description of the *panic-pause* feature in Section 3.2 and OTA capability in Section 3.3, and finally the conclusion of this section in Section 3.4. Note that this module is not covered in the initial product proposal. Its implementation is first suggested by Prof. Timothy Haw-yu Lee during the product’s initial demonstration. Since the robot will actively search for targets, a “panic” button is needed to stop it from causing safety hazards to people (or objects) nearby. In the end, we created a separate section for this component as its implementation is non-trivial.

3.1 Design Verification

Since the robot tracks the user in real-time and fires high-velocity tennis balls, placing a physical panic button on the robot is impractical. Using the existing ESP-12F module, we introduced a Wi-Fi-based control, but the slave ESP is already connected to the thermal camera’s Wi-Fi access point, while the master ESP is emitting a different access point. To resolve this, we implemented ESP-NOW, a protocol by Espressif that enables direct peer-to-peer communication at the MAC layer, bypassing the need for IP addresses or Wi-Fi networks [5]. This allows two ESP modules to communicate even on separate access points. Since our panic button only needs to send a single bit (pause or continue), it comfortably fits within ESP-NOW’s 2000-bit payload limit [6].

3.2 Panic-pause Feature Description

The panic pause feature is implemented by allowing the master ESP to be the access point emitter, and the diagram is shown in Fig. 5. When the machine is booting, the user can connect to this AP and open `http://192.168.4.1`. The user is then presented a home screen with a large green button “TOGGLE PAUSE”. If the robot is active, pressing this button pauses the robot. When the robot is paused, the vertical traverse servo is reset to the level position, the reload servo is moved back to the initial state (refer to Fig. 4a), the horizontal traverse servo is stopped, and the firing motors stop spinning. Additionally,

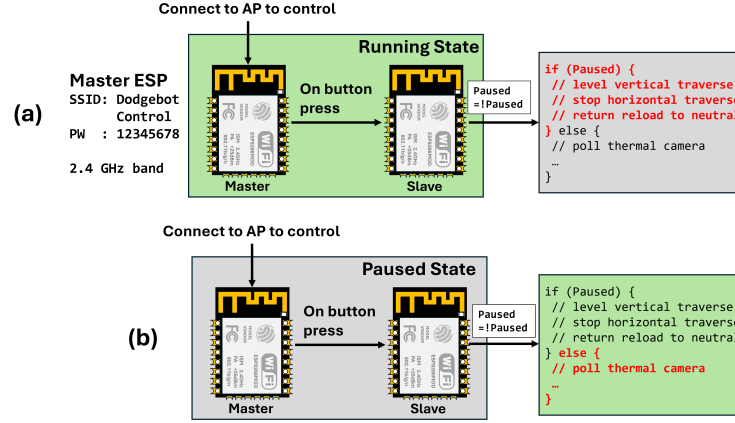


Figure 5: The panic-pause mechanism. When the robot is running (a), the user can connect to the specified SSID and password (can be changed) to open the HTML interface. When the user pressed the "TOGGLE PAUSE" button, the master ESP transmits a one-bit data to the slave ESP, and the slave ESP will update its state. The slave ESP will stop all servos, motors, and image polling. This effectively shuts down the robot. If the user wishes to continue, they may press the button again, which will toggle the slave ESP to continue as normal.

the slave ESP stops polling for new image data from the camera, which also freezes the camera. Pressing this button again will instantly return the robot to its last state, and it will immediately start hunting for targets. This feature allows users to temporarily pause the robot as they wish, which is critical in ensuring that the robot remains safe to use. The code implementation for both master and slave ESP is shown in Appendix D Listing 2.

3.3 On-the-Air (OTA) Flashing Capability

In addition to the panic button, we implemented over-the-air (OTA) firmware updates for the two ESP-12Fs using Arduino's OTA module. This eliminates the need for physical flashing connections like USB-C or FTDI. A separate ESP emits an AP that both ESPs connect to during boot. If the robot is already running, it must be power cycled to enter bootloader mode. Once connected, each ESP is assigned a unique IP, allowing the user to flash them via the Arduino IDE. Importantly, the new firmware must include the OTA setup code (Appendix D, Listing 3); otherwise, future OTA updates will fail. This process is illustrated in Fig. 6.

3.4 Conclusion

In this section we introduced the panic button and the OTA flashing feature. The panic button allows the user to pause the robot at any time, and continue when convenient. The panic button can be toggled using any device capable of Wi-Fi connection, and the interface is made as user-friendly and as simple as possible. This allows the user to toggle the machine with ease without having to understand unnecessary details. The OTA flashing feature allows the delivery of new firmware updates without having to introduce

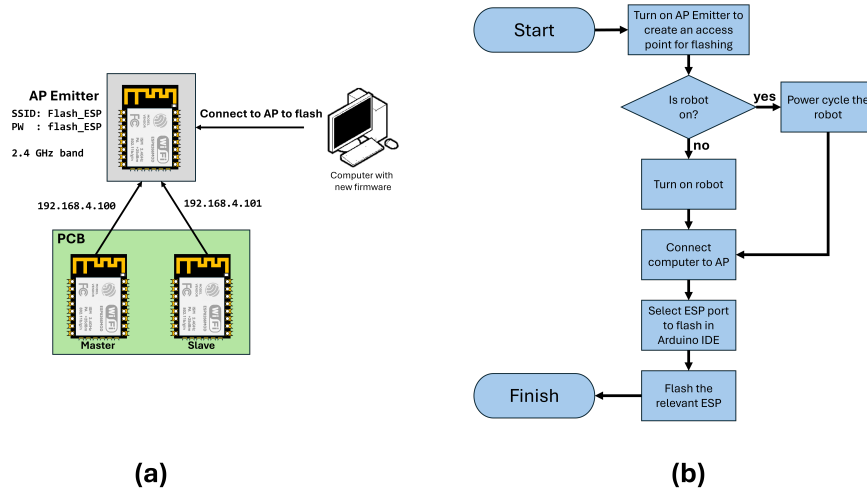


Figure 6: The OTA flashing mechanism. In (a), the two ESPs in the robot connects to an AP Emitter (if available) with a set of SSID and password (can be changed). The computer (which contains the firmware to flash to) connects to this AP. To flash either of the two ESPs, the user can select the appropriate ESP in the Arduino IDE and initiate flash accordingly. (b) shows the flowchart of the OTA flashing process.

wire connections that may create hassle to the user. In the future, we may rent a server that contains firmware updates, and the ESPs can utilize the user’s internet connection to check for new updates, hence removing the need to have a portable AP emitter and possibly remove manual flashing altogether.

4 Machine Vision Module

We now introduce the “brain” of the robot, responsible for controlling the various servos/motors while also being the main processing power behind the object detection algorithm. Section 4.1 describes the machine vision algorithm used to detect objects, Section 4.2 explains how the ESP (i.e., slave ESP) controls the servos and motors using PWM. We also verify that this module is functioning as intended according to the requirements and verification table supplied in Appendix A. Finally, Section 4.4 concludes this section with a brief overview of the machine vision module.

4.1 Object Detection Algorithm Using Trinary Mapping and Weight Classification

The object detection mechanism is comprised of two separate processes: image polling through websocket and centroid detection of the polled image, which will be discussed in Section 4.1.1 and Section 4.1.2 respectively.

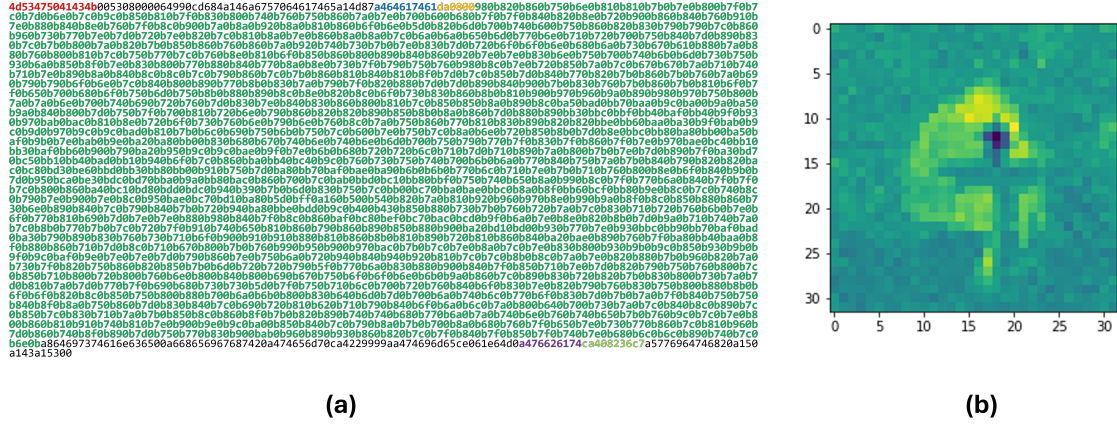


Figure 7: An example of the raw data streamed from the WebSocket protocol (a). Binaries of importance are labeled with colors: red is ASCII for MSGPACK, blue is ASCII for data, yellow signifies the image's binary string length in MessagePack format, dark green is the image binary data, dark purple is ASCII for voltage, and light green is the voltage data of the camera (in float32 format). (b) shows the rendered image of the dark green binary string.

4.1.1 Image Polling Through WebSocket

The thermal camera (RT-Thermal) streams captured images real-time through Wi-Fi (the reason we use the ESP-12F). The data is encoded using MessagePack format [7], and the binaries are streamed using the WebSocket protocol (RFC6455) [8]. We can interface the ESP with the thermal camera by connecting the ESP to the camera's AP and then using Arduino's built-in `ArduinoWebsockets` module to stream WebSocket data. Fig. 7a shows an example of the typical WebSocket payload transmitted from the camera. The image can be obtained by converting the binaries labelled in dark green (see Fig. 7a) into an array of `uint_16s` of length 1024, which can be plotted as a heatmap in Fig. 7b. Therefore, we can shape the 1D array as a 32-by-32, bijective 2D array, with elements $[i, j]$ corresponding to the $[i, j]$ -th pixel. We will now use this 2D array to perform the object detection algorithm.

4.1.2 Centroid Detection Algorithm

We first want to filter images that do not contain blobs (e.g., background heat signature). We can do this by finding the difference between the maximum and minimum value of the image array. If the difference is less than 90, we can assume that the recorded temperature is fairly uniform across all pixels, hence no target is present. If the image contains enough temperature anomaly to exceed this limit, it will then undergo *trinary mapping*. The image array (which is comprised of raw values from 2000 to 3000) is transformed into a trinary image that accepts three values 0, 1, and 2:

$$\text{trinary}[i, j] = \begin{cases} 0 & \text{image}[i, j] < 2970 \\ 1 & 2970 \leq \text{image}[i, j] \leq 2980 \\ 2 & \text{image}[i, j] > 2980 \end{cases}$$

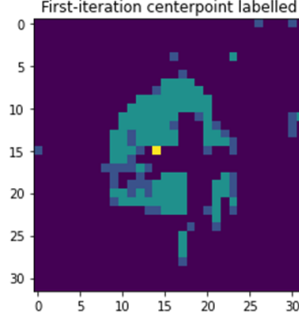


Figure 8: The trinary image along with the first iteration centroid $[i_{\text{mean},1}, j_{\text{mean},1}]$ labelled (yellow).

Now, we can start the detection algorithm. We first calculate the mean value of each rows and columns. A higher mean value indicates a higher presence of the blob in the trinary image. Then, we select each five rows and five columns with the highest mean and compute these averages to obtain $[i_{\text{mean},1}, j_{\text{mean},1}]$, where $i_{\text{mean},1}$, $j_{\text{mean},1}$ indicates the value of the averaged rows and columns of this first iteration respectively. The result of this computation is shown in Fig. 8. Unfortunately, the predicted centroid $[i_{\text{mean},1}, j_{\text{mean},1}]$ may deviate due to contributions from other rows/columns that may contain extraneous blobs. Therefore, to achieve better accuracy we have to filter these problematic points from the final centroid calculation. This is done by first finding the average distance between the centroid and the five selected rows (i_k) or columns (j_k) from earlier to obtain the average distance \bar{d}_r and \bar{d}_c , shown in (2). We will use this value to filter out rows or columns that lie further than \bar{d}_r and \bar{d}_c respectively.

$$\bar{d}_r = \sum_{k=0}^4 \frac{1}{5} |i_{\text{mean},1} - i_k| \quad (1)$$

$$\bar{d}_c = \sum_{k=0}^4 \frac{1}{5} |j_{\text{mean},1} - j_k| \quad (2)$$

After filtering, we calculate again the predicted centroid to get $[i_{\text{mean},2}, j_{\text{mean},2}]$, i.e., the final centroid. This is shown in Fig. 9, which is an improvement over Fig. 8 as now the predicted centroid lies within the blob region. Finally, we determine the offset $[i_{\text{mean},2} - 16, j_{\text{mean},2} - 16]$ to find the amount of deviation the centroid has from the center of the image. This value is then processed by the ESP into values meaningful for the servos and motors.

4.2 Motor and Servo Control

The pixel offset calculation outlined in Section 4.1.2 is not enough to determine the angle needed to rotate the gun. Since the servos operate with PWM pulses, we need to determine the angle that causes the pixel in the camera to drift by one pixel, and then find the appropriate instruction values to rotate the gun by this amount. We obtained that an

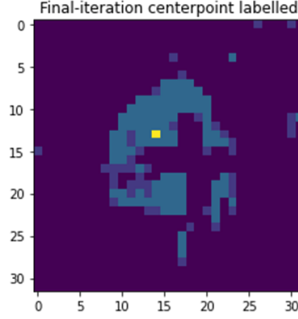


Figure 9: The trinary image along with the first iteration centroid $[i_{\text{mean},2}, j_{\text{mean},2}]$ labelled (yellow). The centroid now lies within the blob.

offset of one pixel corresponds to 1.5° , hence we can assume that a shift of n pixels must correspond to $n \times 1.5^\circ$. Finally, the servos RDS3225 (vertical and horizontal traverse) and MGR996R (reload) accepts pulse widths 1000 to 2000 μs [9], [10]. Using this information, we can derive a table that corresponds to the pulse width in microseconds, shown in Table. 1.

Servos	Stop	CCW	CW
Horizontal	1500	1800 ($6^\circ/s$)	1200 ($6^\circ/s$)

(a)

Servos	Level	Maximum	Minimum	Increment
Vertical	1250	1500 (up)	1000 (down)	+25/ 10°
Reload	1500	2000 (back)	1100 (front)	+25/ 10°

(b)

Table 1: Tabulation of PWM widths (in microseconds) for (a) horizontal traverse and (b) vertical traverse and reload servo. Since the horizontal traverse servo is a continuously-rotating 360° variant of the RDS3225, 1500 corresponds to stopping the servo, > 1500 rotates the servo counterclockwise, and < 1500 rotates the servo clockwise. The more the value deviates from 1500, the faster it rotates.

To move the servos, we will utilize Arduino's `Servo` module to write PWM signals to the servos and motors. If the centroid is located within two pixels from the center, then the gun will not correct its position. For targets that is located more than two pixels away, we implemented a P -controller along with velocity calculation to determine how much the servos should move. Velocity $v_{i,j}$ is calculated by finding the displacement between the last pixel and the current pixel $v_{i,j}$, and the predicted position is determined to be the position of the blob after 50 milliseconds (see (3)). We then apply a proportional controller $K_H = 15$ and $K_V = 9$ for horizontal and vertical traverse respectively to obtain $K_{H,V} \times \text{pred}_{i,j}$. From our tabulated values in Table. 1, we limit this value to be within

± 300 for horizontal traverse, and ± 200 for vertical traverse. This will give `hspeed` and `vspeed` respectively.

$$\text{predicted}_{i,j} = \text{current}_{i,j} + v_{i,j} \times 50 \text{ ms} \quad (3)$$

Finally, we can write the pulse width needed to rotate the two servos, shown in Listing. 1. By using a P-controller, the turret will rotate smoother (compared to discrete jumps between maximum speed and stationary) while using the velocity calculation we can roughly predict the blob's position and lead our aim accordingly. Note that a different PWM mechanism is used to control the firing motors. This will be elaborated in Section 5.

Listing 1: PWM Control of horizontal and vertical traverse.

```
hTraverse.writeMicroseconds(H_STOP - hspeed);
vTraverse.writeMicroseconds(V_LEVEL - vspeed);
```

4.3 Design Verification

We first tested this module by placing a target at a distance of 5 meters. We noted that the robot is able to detect reliably even at 6 meters, while it started to show issues in identifying the target at about 7 meters or more. We modify requirement No.3 to also include the whole process of polling the image up to relaying PWM signals to appropriate servos to demonstrate how fast our mechanism is. During runtime, we observed that it typically takes about 20-50 ms to read image from the camera, compute the centroid, and determine the PWM signals. This is significantly faster than what is set in the requirements.

Note that requirement No.4 is not implemented in the design. From our experience, programming the turret to rotate clockwise if no target is seen can cause problems especially if the user is in the far-end of the robot. Therefore, it is more logical for the user to bait the robot to aim at them instead of waiting for the robot to reacquire the user. Furthermore, we also implemented velocity calculation, which is able to catch fast targets at 5 meters. The camera tend to overcompensates at further distance since the centroid calculation becomes erratic due to the decreased presence of the target's heat signature. Nevertheless, our velocity prediction can compensate fast moving targets, even with the relatively low camera resolution (32-by-32).

4.4 Conclusion

In this section, we discussed how the ESP collects data from the thermal camera, identifies the centroid of a blob (if it exists), and power the servos/motors accordingly using a combination of proportional control and target estimation using velocity calculation. We have successfully verified our algorithm in a variety of conditions, and our robot is able to achieve good tracking even with fast moving targets. We also noted that our algorithm is very compact, as it can entirely fit in the 32 KiB instruction memory of the ESP-12F, while still leaving enough room for additional OTA mechanism and other smaller programs.

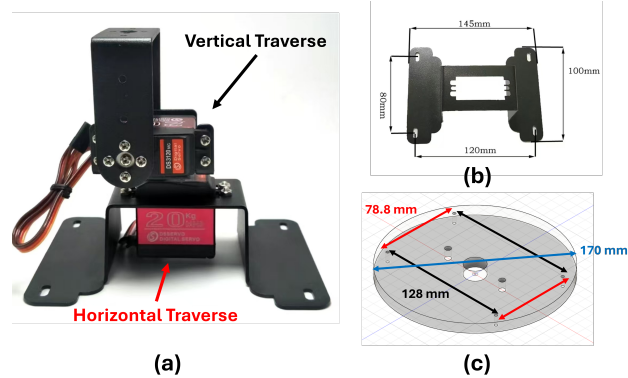


Figure 11: The gun platform (a) and the dimensions of the supporting leg (b). (c) shows the acrylic mount used to hold the platform in (a) on the robot's base. Detailed schematics are supplied in Appendix G.

PWM pin (Fig. 10c), connecting it to ESP's IO12 pin configured as an input to prevent damage from backflow (since ESP pins are 3.3V). The AK300S accepts PWM frequencies between 10 Hz and 2 kHz; we chose 1 kHz via `analogWriteFreq(1000)` and adjusted motor speed through duty cycle (0-100%). This approach offers smooth speed control and greatly reduces back-EMF issues.

5.2 Structural Design of the Gun and Magazine System

We will now discuss the structural design that makes the turret possible. The turret is comprised of three major components: the **platform**, which holds the gun, the magazine, along with the motors and servos together, the **gun**, which forms the structure of the main firing mechanism, and the **magazine**, which holds the tennis balls.

5.2.1 Gun Platform

The platform used is shown in Fig. 11a. It houses the horizontal traverse servo (red) and the vertical traverse servo (black). This platform is mounted on top of a 1 cm-thick acrylic board, shown in Fig. 11c. We choose acrylic as the base to prevent ESD shocks that may happen in the unlikely event of the power supply failing. The acrylic is made 1 cm thick to ensure that it is sturdy enough to hold the gun without bending.

5.2.2 Gun Module

The gun is comprised of a 25 cm steel bar that holds two 48 V DC motors together. Each DC motor mounts a steel wheel with rubber padding on the outer surface of the wheel. The rubber acts as a damper to prevent the ball from getting sliced as it pass through the gun chamber (this actually happened before!). The gap is narrow enough for the ball to pass through unimpeded, but still large enough to ensure that the ball can still

before shutting down to prevent overvoltage. In any case, it is not important to identify the exact amount of back-EMF produced since the PWM solution is a clearly better solution

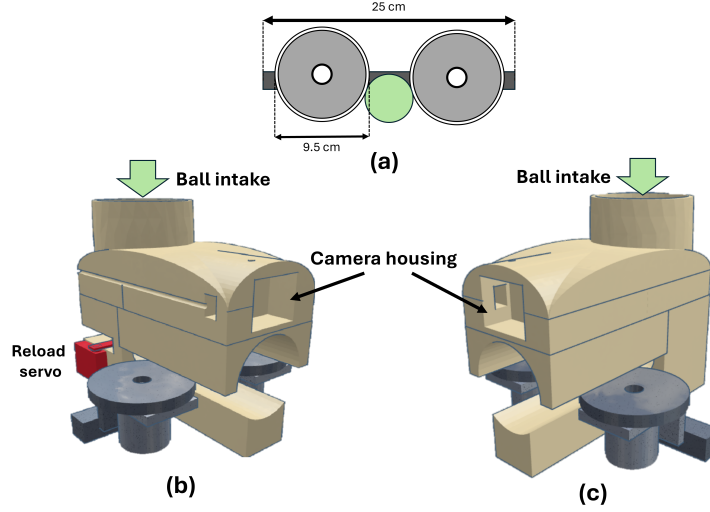


Figure 12: The gun dimensions (a), along with its housing, shown as a 3D render in (b) and (c), each showing the right and left side of the whole gun structure accordingly. For (b) and (c), components colored in gray is the gun while components in yellow is the gun chamber. Additionally, (b) also shows the reload servo, which has a spoke that rotates in and out during the reloading process. Detailed schematics are supplied in Appendix G

be removed from the chamber reliably. We also built the gun chamber which acts as the ball's pathway from the magazine to the firing motors. It also houses the reload servo and the thermal camera, as shown in Fig. 12.

5.2.3 Magazine

To create a fun experience for the user, the robot must be able to hold enough balls so that the user will not tire out in having to refill the magazine constantly. We experimented with an open-top semicircle design, where it functions similarly to a funnel that helps select balls to enter the gun chamber one-by-one. However, this design is prone to jamming; as two balls try to fit through the funnel's smaller end, the balls will try to block each other, causing the reload mechanism to fail. We may implement a straight tube that houses the ball, but this design is inefficient with space, and can lead to the final product being too awkwardly tall. Therefore, we employ a spiral shape for our magazine design. It is comprised of three parts, the bottom connector that connects the gun chamber with the spiral, and the top connector that allows users to load new balls into the spiral magazine. Fig. 13a shows the 3D render of the spiral, and Fig. 13b-d shows the three separate components that make up the complete magazine subsection.

5.3 Design Verification

Our testing shows that the firing motor can consistently launch balls beyond the 5 meters range set in Appendix B. However, we choose to not implement requirement No.5, i.e., to stop the firing motor when a target is not found. Our justification is that typically the

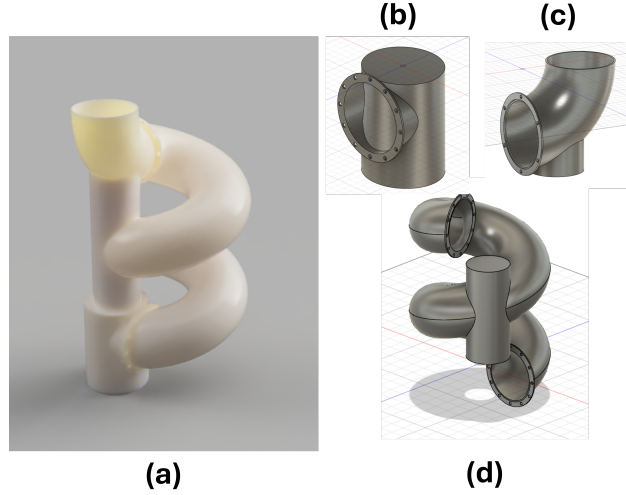


Figure 13: The spiral 3D render (a), which is comprised of three parts. (b) is the bottom connector that connects to the gun chamber, (c) is the top connector that allows users to replenish the magazine, and (d) is the spiral that acts as the tennis ball magazine. Detailed schematics are supplied in Appendix G.

robot is used in a high-activity environment, which means that the time window of the target not in view of the robot is narrow. This is reflected during testing, as the robot will continuously turn off and turn on the firing motors, which will degrade the motors and the motor drivers. Triggering the on/off of the motor may also damage the power supply, as we will be accumulating back-EMF while simultaneously inducing frequent current spikes on the system (which happens when the motor is first started). Our reload motor is sufficient in ensuring no stray tennis balls enter the gun chamber, hence we do not find the need to make an automatic trigger for the firing motors.

5.4 Conclusion

In this section, we detailed the design and implementation of the *dodgebot*'s firing system. We demonstrated how the firing motors can be controlled using PWM signals from the ESP. We employ space-saving measures in building the gun platform and the magazine. Specifically for the magazine, we utilize an innovative spiral design to allow more balls to be loaded into the machine while keeping the overall robot's height as low as possible. We verified that this system meets our tests as outlined in the requirements and verification table.

6 Power Module

In an effort to achieve a production-level *dodgebot*, we implemented the power module to handle the distribution of power from 230 AC (commonly used in China) to appropriate DC voltages required by our electronics. We will introduce how the power distribution

Table 2: List of load components that needs to be powered.

Component	Voltage	Current Draw	Power Estimate
2× 48 V motors	48 V	5.2 A	250 W
2× RDS3225	6 V	2 A	18 W
MG996R	6 V	1 A	3 W
Thermal Camera	5 V	1 A	5 W
2× ESP-12F	3.3 V	400 mA	1.32 W
Total		9.6 A	277 W

system works, and verify our implementation as written from the requirements and verification table in Appendix C.

6.1 Power Distribution Using Step-Down Converters

Power distribution is first done using an AC to 48 V DC converter (S-500-48). The S-500-48 can provide up to 10.4 A before its overcurrent protection mechanism turns on. It also has a built-in fan to dissipate heat generated from the conversion process. The converter is directly connected to two 48 V motors, each drawing current up to 2.6 A. To power other modules, we need to implement another step-down converter that converts 48 V to 6 V, which will be used to power the servos. Additionally, we will have another converter that converts 48 V to 5 V USB, which is used to charge the thermal camera if needed. Finally, we would like to power the two ESP-12Fs. These two devices require 3.3 V to operate, hence another step-down converter is required (AMS1117-3.3) to convert 6 V to 3.3 V. Table 2 shows the power requirement for each modules, with current listed as its maximum current rating. It is worth nothing that in normal operations, the current draw is much lower: each 48 V motors draw about 300 mA, the combined servos rarely, if ever, exceed 1 A during normal operations, and the thermal camera has its own built-in battery that can last up to two hours. Therefore, we concluded that the S-500-48 converter is safe to use in our design. Fig. 14 shows the complete power distribution used in this robot design.

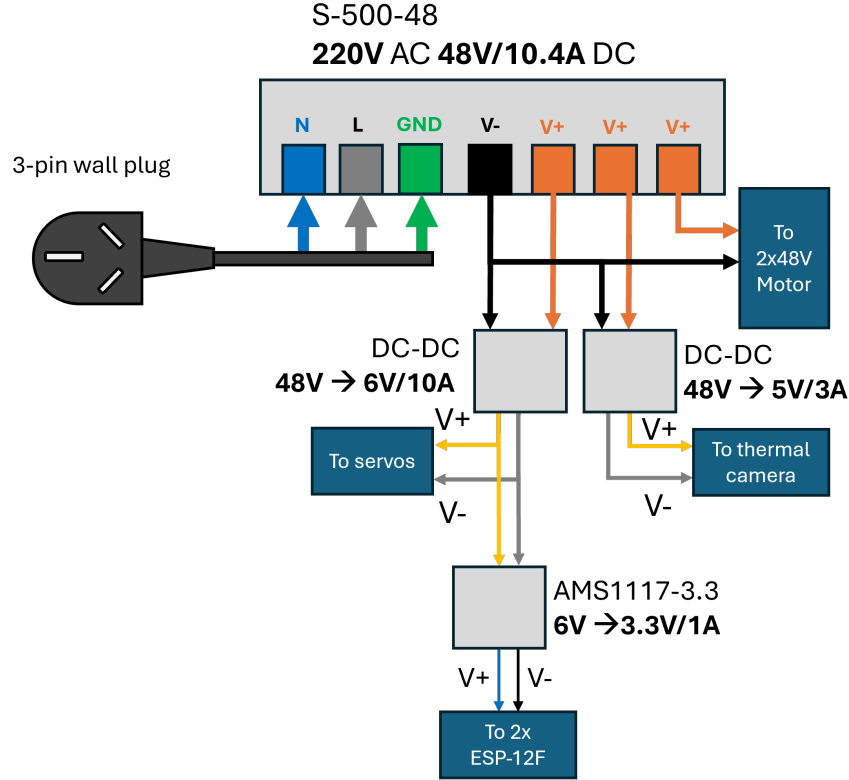


Figure 14: The power distribution diagram. The S-500-48 AC-DC converter has 3 V- pins, which is represented as one V- pin to improve visibility. Gray blocks indicate converters, while dark blue blocks indicate load components.

6.2 PCB Design for ESP-12F and AMS1117

Our robot contains a PCB component to house the AMS1117-3.3 voltage converter, two ESP-12Fs, and PWM pins to control servos/motors. When designing the PCB, we first take into consideration the circuit recommendation of AMS1117-3.3 and ESP-12F. The former requires one input ($10\ \mu F$) and one output capacitor ($22\ \mu F$) for stability [13], while the latter requires a $10\ k\Omega$ pull-down resistor in IO15, and several $10\ k\Omega$ pull-up resistors for normal operation i.e., this circuit will not bring the ESP into bootloader mode. This does not pose any problem as our ESPs are flashed using OTA (see Section 3.3). The PCB design and render is shown in Fig. 15 (schematic is attached on Appendix F). Note that both ESP's antenna hangs from the PCB, in accordance to Espressif's *PCB Design and Module Placement Guide* [14].

In addition to the 6V power pins and H3 header pin, there are also extra header pins labeled H4 and H5. These are the pins that connect to the PWM and CONTROL 1 pins of the AK300S drivers. The "+5V" pin from AK300S connects to either "+5V1" or "+5V2" and outputs "COM1+" or "COM2+". The "PWM1" and "PWM2" pins connect to the IO13 and IO12 pins in the slave ESP respectively. The former has an additional $470\ \Omega$ load resistor, since according to [11] the output current of the "+5V" pin is about 20 mA. If we assume that the optocoupler LED (see Fig. 10c) has forward voltage 1.2 V, then we obtain

the voltage across a resistor R (that connects "+5V" with "COM+") to be 3.8 V. Since the optocoupler circuit used in AK300S is unknown, we choose $R = 470\ \Omega$ as to keep the current within safe limits (5-20 mA), which gives 8 mA. From testing, 8 mA is enough to trigger the optocoupler, hence we set $R = 470\ \Omega$ for the final product. We also introduced a $150\ \Omega$ resistor between the PWM pin and the ESP's IO pin as an additional precaution to current/voltage spikes from the optocoupler input.

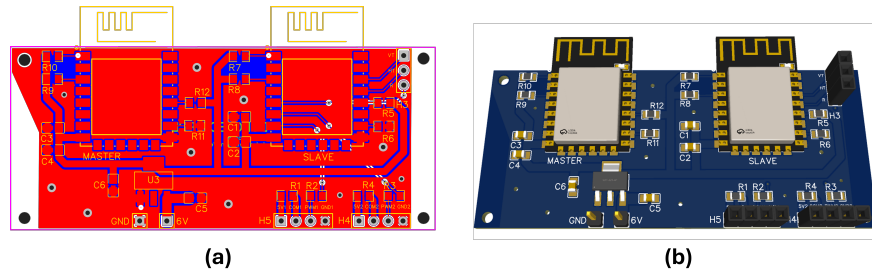


Figure 15: PCB schematic (a) and the 3D render (b). For (a), red represents the top layer, blue represents the bottom layer, and orange represents the top silk layer. There are four screwing holes on each of the board edges. Note that the ESP antenna hangs from the PCB as required from the datasheet.

6.3 Design Verification

Our final design is different than what we originally intended when building the R&V table for this module. To elaborate, requirement No.2 requires the delivery of 24 V/3 A to the horizontal motor. In our final design we opted for a 360-continuous rotating servo instead, which draws 6 V/2 A. We choose this servo so that we are only required to provide 48 V, 6 V, 5 V, and 3.3 V instead of an additional 24 V supply. Our 6 V DC converter can withstand up to 10 A, similarly with our AC-DC converter. Since the total current from the 6 V supplies are less than 5 A, we are able to power three servos simultaneously. However, we note that the output voltage of this DC converter is about 6.2 V. Fortunately, the servos MG996R and RDS3225 have maximum voltage of 6.6 and 6.8 V respectively, and the AMS1117 can receive a maximum of 18 V.

We also examined the 3.3 V conversion process. Our initial tests show that the AMS1117 is capable of converting 6 V, but the output ranges between 3.1-3.2 V. Nonetheless, our testing shows that it can reliably supply beyond 400 mA for two ESPs for more than an hour. The 48 V to 5 V converter in reality outputs 5.1 V. This still falls within the allowable range of the USB 2.0 specification (4.75-5.25 V) [15]. In conclusion, we are still able to show that the components described in this section functions as intended.

6.4 Conclusion

We have introduced the power module of *dodgebot* in this section. We explored how power is distributed among the load devices, and how we built a PCB board to interface the ESP-12F MCU with the AMS1117-3.3 voltage converter. We also introduced

Table 3: Labor cost for each member in dollars. The compensation is calculated using the formula outlined in [16].

Name	Work hours	Compensation Required
Loigen Sodian	428	21,400
Jaden Peterson Wen	428	21,400
Qingyan Li	332	16,600
Putu Evaita Jnani	214	10,700
Isaac Koo Hern En	55	2,675

7 Cost Estimation

7.1 Expenses on Parts

The total expense required to build the robot is attached in Appendix E. We achieved a total expense of 2242 RMB, which is 1242 RMB over the initial 1000 RMB budget. We attribute this high cost to some components such as the 48 V motors (500 RMB), the thermal camera (299 RMB), and the cost of printing the spiral tube (200 RMB). Additionally, we realize that communication between team members needs to be improved, and that a more careful planning (especially when concerning dimensions) is crucial to ensure that no modules bought will turn into waste. There are a few instances where miscommunication like this happen, which may also contribute to the high development cost. Nevertheless, we believe that future iteration of this robot can be done at a cheaper price of 1800 RMB. We can also replace the 48 V motor with a 24 V motor, which may cut down the price significantly, as we can use a cheaper BLDC driver, and also allows the downsizing of the AC-DC converter to the S-400-24 model, which is 20 RMB cheaper. However this requires further testing to ensure that the firing range is not compromised.

7.2 Labor Reimbursements

Using the formula provided in the *ECE 445 Final Report Guidelines* [16], Table 3 shows the labor cost of each members in the group assuming an hourly wage of \$20, calculated from February 1, 2025 to May 18, 2025.

8 Conclusion

We successfully built a dodgeball machine capable of throwing tennis-sized balls at a minimum distance of 5 meters. We achieved this by using two 48 V, 6000 rpm motors with respectable torque output (0.15 Nm). To identify targets, we use a thermal camera

to read information from the surroundings and process each images real-time to obtain the centroid of the target (if it exists). This centroid data is then translated into PWM signals that control the servos responsible for rotational movement of the gun. The gun adopts a *spiral magazine* system that saves space, efficient in carrying many balls, and is aesthetically pleasant. The gun has a reload servo that ensures the reload process is controlled, and that projectiles are correctly pushed into the chamber.

We also conducted production feasibility from an economical perspective. While we believe that improvements must be made in the planning stages of the project, the process that leads up to this design is a series of repeated failed experiments in prototyping the final implementation of the robot, which we attribute as part of a normal development cycle. We believe successive iteration of the product should bring the price down to the 1700 to 1800 RMB range. The design is built as closely as possible to production-level quality, but we may show future improvements especially in the aesthetic design of the robot, alternative components that may be cheaper (e.g., replacing the 48 V motor) or better (e.g., replacing two ESP-12Fs with one ESP-32). Nevertheless, we believe that this work has good potential to be marketed as a polished product in the future.

References

- [1] mealeap Inc., *Hado - techno sports*, <https://hado-official.com/en/>, Accessed: 2025-05-16, 2025.
- [2] Kandidat, *Dodgeball object detection dataset and pre-trained model*, <https://universe.roboflow.com/kandidat-pgnqx/dodgeball-f7dp5>, Accessed: 2025-05-16, Apr. 2023.
- [3] A. Weber, M. Wilhelm, and J. Schmitt, "Analysis of factors influencing the precision of body tracking outcomes in industrial gesture control," *Sensors*, vol. 24, no. 18, 2024, ISSN: 1424-8220. DOI: 10.3390/s24185919. [Online]. Available: <https://www.mdpi.com/1424-8220/24/18/5919>.
- [4] Ai-Thinker Technology Co., Ltd., *ESP-12F 802.11 b/g/n Wi-Fi Module Product Specification*, https://docs.ai-thinker.com/_media/esp8266/docs/esp-12f_product_specification_en.pdf, Accessed: 2025-05-16, 2017.
- [5] Espressif Systems, *Esp-now wireless communication protocol*, <https://www.espressif.com/en/solutions/low-power-solutions/esp-now>, Accessed: 2025-05-16.
- [6] E. Systems, *Esp8266 arduino core - espnow.cpp source code*, <https://github.com/espressif/esp-now/blob/master>, Accessed: 2025-05-17, n.d.
- [7] MessagePack Contributors, *Messagepack: It's like json. but fast and small*. Accessed: 2025-05-17, 2025. [Online]. Available: <https://msgpack.org>.
- [8] I. Fette and A. Melnikov, *The websocket protocol*, RFC 6455, Accessed: 2025-05-17, Dec. 2011. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc6455>.
- [9] DS Servo, *Rds3225 servo motor datasheet*, <https://www.cytron.io/p-rds3225>, Accessed: 2025-05-17, n.d.
- [10] Tower Pro, *Mg996r servo motor datasheet*, https://www.ee.ic.ac.uk/pcheung/teaching/DE1.EE/stores/sg90_datasheet.pdf, Accessed: 2025-05-17, n.d.
- [11] AZK Motor, *Ak300s bldc motor driver product page*, <http://www.azkmotor.com/ProductDetail/8679184.html>, Accessed: 2025-05-17, n.d.
- [12] Microchip Technology, *Mcp41010 digital potentiometer datasheet*, <https://www.microchip.com/en-us/product/mcp41010>, Accessed: 2025-05-17, n.d.
- [13] Advanced Monolithic Systems, *AMS1117 Low Dropout Voltage Regulator Datasheet*, <https://www.diodes.com/assets/Datasheets/AMS1117.pdf>, Accessed: 2025-05-17, 2004.
- [14] E. Systems, *ESP-WROOM-02 PCB Design and Module Placement Guide*, https://www.espressif.com/sites/default/files/documentation/esp-wroom-02_pcb_design_and_module_placement_guide_0.pdf, Accessed: 2025-05-18, 2016.
- [15] USB Implementers Forum, *Universal serial bus specification revision 2.0*, <https://www.usb.org/document-library/usb-20-specification>, Section 7.2.1: Power Distribution Overview, Apr. 2000.
- [16] ECE 445 Course Staff, *ECE 445 Final Report Guidelines*, <https://courses.grainger.illinois.edu/ece445zjui/documents/ECE.445.Final.Report.Guidelines.pdf>, Accessed: 2025-05-18, 2024.

Appendix A Machine Vision R&V Table

Table 4 shows the *requirements and verification* table for the machine vision module.

Table 4: Requirements and Verification Table for Machine Vision Module.

Subject Category	Requirements	Design Verifications
Image Classification and Targeting	1) When the absolute difference between a pixel and its neighboring pixels exceeds 105, the system should flag the image as a candidate thermal anomaly and initiate blob detection.	<p>a) Set the computer to output “Candidate Anomaly” when a candidate anomaly is marked.</p> <p>b) Input a matrix of 32-by-32 pixels into the computer, with the center pixel at a typical human temperature and neighboring pixels at ambient temperature.</p> <p>c) Record whether the system outputs “Candidate Anomaly”.</p> <p>d) Enter a 32 x 32 pixel matrix into the computer with all pixels at 25°C, this can be done by placing the camera lens directly on top of someone’s arm for example.</p> <p>e) Record whether the system outputs a “Candidate Anomaly”.</p> <p>f) Perform at least 3 validations per group.</p>
	2) Within a distance of 5 meters, a thermal anomaly pixel is classified as valid when the MCU returns the centroid position.	<p>a) Prepare a tester to stand within five meters of the system and take at least five sets of images.</p> <p>b) Prepare a piece of paper similar in shape to the tester to be fixed within five meters from the system and take at least five sets of images.</p> <p>c) Record the number of times the system identifies the object.</p> <p>d) The F1-score $2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$ of the system for the classification results should be greater than 80% as a composite metric.</p>

Table 4 – continued from previous page

Subject Category	Requirements	Design Verifications
	3) Once a valid pixel is detected, the system should transmit a command containing the target coordinates to the Arduino within 1 second.	<p>a) In the online environment, set the computer output the time at this point (T_1) after sending coordinate.</p> <p>b) Input an image with a valid object into the MCU.</p> <p>c) Use an oscilloscope to monitor the turret motor output voltage, with the oscilloscope input and output connected to both ends of the motor. When the motor output voltage is greater than 5 V, record the time at this point (T_2).</p> <p>d) Make at least three measurements. The average time difference $\langle T_1 - T_2 \rangle$ should be less than 1 second.</p>
	4) If no target is detected, the system should rotate the turret clockwise by $30^\circ (\pm 2^\circ)$ and then initiate a new classification cycle within 1 s.	<p>a) Connect the rotary potentiometer to the turret motor so that the motor rotation can drive the potentiometer knob. The three pins of the potentiometer are connected to VCC ($24\text{ V} \pm 3\text{ V}$), Arduino analog input (A0) and GND (less than 1 V).</p> <p>b) Input a background image (i.e., no targets). The motor should rotate clockwise.</p> <p>c) Use the Arduino to read the analog value (0-1023) via <code>analogRead()</code>, map it to an angle (0°-300°), and record it.</p> <p>d) Take at least three measurements. The rotation angle should be between 28° and 32°.</p>

Appendix B Firing System R&V Table

Table 5 shows the *requirements and verification* table for the firing system module. Note that in the final design the turret and the firing system is integrated into one module. Hence Table 5 shows the R&V table for both the turret and firing system module.

Table 5: Requirements and Verification Table for Firing System Module.

Subject Category	Requirements	Design Verifications
Gun Rotational Adjustment	1) The vertical traverse servo should be able to aim up to 20 degrees, and it should hold this position for at least 20 seconds while powered.	a) Program the ESP module to lift the turret by 20 degrees. b) Measure and confirm the 20° inclination. c) Let the servo hold this position for 20 seconds. Test is successful if it maintains this.
	2) The vertical traverse servo should aim up according to the tabulated values.	a) Program the ESP to aim by 1–16 pixels. b) Servo should raise to the corresponding angle. c) Assume target is at a 5-meter distance.
	3) The vertical traverse servo should return to neutral (0°) after firing within 1 second.	a) Mock reload motor signal to ESP. b) ESP must return servo to 0° within 1 second (check via LED or Serial Monitor). c) Confirm servo angle is zero.
	4) The horizontal traverse motor should rotate 360° within 20 seconds.	a) Program ESP to rotate motor 360° (both directions). b) Time begins when LED or Serial Monitor signals movement. c) Verify full rotation within 20 seconds.
	5) The horizontal motor should turn according to tabulated values.	a) Program ESP to move 1–16 pixels left and right. b) Motor should rotate according to table.

Table 5 – continued from previous page

Subject Category	Requirements	Design Verifications
Projectile Reload Mechanism		c) Assume target is at a 5-meter distance.
	6) When no target is detected, the turret should rotate 30° clockwise.	a) Load image with no target. b) ESP rotates turret 30° clockwise. c) Confirm 30° rotation.
	1) Reload motor triggers only when blob detector detects an object.	a) Load image with/without target. b) If target is identified, reload motor should activate. c) Test passes if ESP behaves as instructed regardless of detection accuracy.
	2) Motor loads one ball at a time; fires 5 in 10s; 1 ball every 2s; then halts until new target.	a) Load image with valid target. b) Start timer when LED activates after detection. c) Record time to load and fire each ball. d) Confirm 5 shots, then system sleeps for 5s and reload motor stops.
	3) Reload mechanism fails less than 10% of the time.	a) Load two valid target images. b) After test, system should misfire at most once.
	4) Firing motor should launch ball at least 5 meters, 80% of the time.	a) Launch 10 balls, record each launch. b) Measure shortest distance from launch point to landing.
	5) Firing motor only triggers when valid object is detected.	a) Load image with/without target. b) Motor should only activate on detection. c) Test passes if ESP follows logic regardless of actual accuracy.

Appendix C Machine Vision R&V Table

Table 6 shows the *requirements and verification* table for the power module.

Table 6: Requirements and Verification Table for the Power Module.

Subject Category	Requirements	Design Verifications
Power Delivery	1) Delivers 48 V and 3 A max to the firing motor continuously. Note that the component will still draw power even when idle.	We verify that when plugged in, the motor receives about 48 V, but near zero current (idle). When the motor is activated, it should maintain 48 V (and not drop).
	2) Delivers 24 V and 3 A max to the horizontal traverse motor continuously. Note that the component will still draw power even when idle.	We verify that when plugged in, the motor receives about 24 V, but near zero current (idle). When the motor is activated, it should maintain 24 V (and not drop).
	3) 3) Delivers 6 V and 2 A max to the vertical traverse servo and reload motor continuously. Note that the component will still draw power even when idle.	We verify that when plugged in, the motor(s) receives about 24 V, but near zero current (idle). When the motor is activated, it should maintain 24 V (and not drop).
	4) Delivers 3.3 V 200 mA max to the ESP module.	We verify that when plugged in, the ESP receives about 3.2-3.3 V, and that the current does not exceed 200 mA.

Appendix D Code Listings for Operation Module

Listing 2: Implementation of OTA in the master/slave ESP.

```
#include <espnow.h> //for both master and slave ESP

/* Master ESP */
uint8_t slaveAddress[] = {0x08, 0xF9, 0xE0, 0x6C, 0x36, 0xB1};

void setup() {
    WiFi.mode(WIFI_AP_STA);
    WiFi.softAP(ssid, password); //ssid and password as shown in this
    section

    esp_now_set_self_role(ESP_NOW_ROLE_CONTROLLER);
    esp_now_register_send_cb(OnDataSent);
    esp_now_add_peer(slaveAddress, ESP_NOW_ROLE_SLAVE, 1, NULL, 0);

    server.on("/", HTTP_GET, handleRoot);
    server.on("/send", HTTP_GET, handleSend);
}

/* Slave ESP */
bool Pause = false;
void onReceive(uint8_t *mac, uint8_t *data, uint8_t len) {
    if (len == 1 && data[0] == 1) {
        Pause = !Pause; // Toggle the pause flag
        Serial.printf("Pause toggle: now %s\n", Pause ? "ON" : "OFF");
    }
}

void setup() {
    esp_now_set_self_role(ESP_NOW_ROLE_SLAVE);
    esp_now_register_rcv_cb(onReceive);
}

void loop() {
    if (Pause) { //pause the device }
    else { continue as normal }
```

Listing 3: Implementation of OTA in the master/slave ESP.

```
#include <WiFiUdp.h>
#include <ArduinoOTA.h>

bool otaEnabled = false;

void setup() {
```

```
WiFi.mode(WIFI_STA);
WiFi.begin(OTA_SSID, OTA_Password);

if (WiFi.status() == WL_CONNECTED) {
    otaEnabled = true;
    ArduinoOTA.onStart([]() { ... });
    ArduinoOTA.onEnd([]() { ... });
    ArduinoOTA.onProgress([]() { ... });
    ArduinoOTA.onError([]() { ... });
} else { //run robot normally }
}

void loop() {
    if (otaEnabled) {
        ArduinoOTA.handle();
    } else { //run robot normally }
}
```

Appendix E Detailed Expenses

Table 7 shows the total expenses during the development of *dodgebot*.

Table 7: Total expenses for *dodgebot*.

No.	Item Name	Quantity	Price (¥)
1	Pocket-size Thermal Imaging Device (Black)	1	299
2	Star-Shaped Coupling D25 L30	2	24
3	Heavy Duty Iron Caster Wheel (4-inch, 12 holes)	2	22
4	Hex Socket Bolt M22×50 (8.8 grade, 2 pieces)	1	6.24
5	Plum Coupling Star Type D40 L45	2	40
6	Plum Coupling Star Type D45 L55	1	30
7	Aluminum Profile 4040 - 2020L Type	1	16.19
8	T-slot Screw + M5 Nut Set	1	1.6
9	Spring Plate Nut for Aluminum Profile (20 pcs)	1	2.4
10	2020 Corner Brackets for Aluminum Profile	1	0.58
11	Hinged Connector (Zinc Alloy, 20 Series Slot)	1	0.6
12	Stepper Motor Mount Bracket for 42/57/60/80 size motors	2	36
13	Plum Coupling Star Type D45 L55 (duplicate)	1	30
14	EVA Foam Balls (Photography Prop - 6 pcs)	1	11.8
15	4-inch Silent Swivel Rubber Wheel (TPR)	2	37.2
16	Brushless DC Motor Kit 57BLY55-48V-100W	2	493
17	UPVC Pipe + 90° Elbow (DN80, 90mm diameter)	2	27.19
18	MG996R Metal Gear Servo Motor (360° rotation)	1	14.81
19	Dual-axis Servo Motor Set 25kg (RDS3225)	1	105
20	Drive Wheel (95mm diameter, 8mm inner)	2	54
21	Acrylic Transparent Dome Cover (200mm)	1	31
22	Stainless Steel Coupler (Double-hole, 3-6mm)	1	7.84
23	Shaft Collars with Hex Set (10pcs + 5 Handle Tools)	2	11.07
24	Short Solid Stainless Steel Rod ($\varnothing 3 \times 200$ mm, 5pcs)	1	2.67
25	ergo5400 Glue	1	12.9

No.	Item Name	Quantity	Price (¥)
26	30KG DS3230 Servo Motor	1	97
27	2DOF Metal Gimbal Bracket	1	37
28	36V48V60V to 6V5A Converter	1	45
29	25T Stainless Steel Servo Disc	1	43.5
30	5A Magnetic Charging Cable (2m, Type-C)	1	13.9
31	M5*10 Hex Screws (50 pcs)	1	2.24
32	20M5*10T Slot Screws + M5 Flange (50 sets)	2	5.3
33	2020 Plate Angle Brackets (M5 Screws x4) (28 pcs)	1	1.9
34	M5 Lock Nuts (50 pcs)	1	10
35	Aluminum Profile 2020E (1.5mm)	1	61
36	Q235 Steel Plate Laser Cut (Custom Size)	1	89
37	Transparent Acrylic Sheet A4 210x297x2mm	1	28
38	Transparent Acrylic Sheet 800x800x2mm (2 pcs)	2	92.3
39	Heater Support Stand	1	15.52
40	Turntable Base 250mm	1	28.8
41	M3*4 Screws (50 pcs)	1	7.5
42	Transparent Acrylic Sheet (Custom Blue)	1	28
43	Valigo V-863 Acrylic Glue + Syringe	1	11.57
44	MCP41010-I/SN Digital Potentiometer (6 pcs)	6	70.4
45	ESP8266 NodeMCU WiFi Module	1	26
46	Custom 3D Printing Structure	7	199.5
47	Acrylic Transparent	1	11
Total		69	2241.52

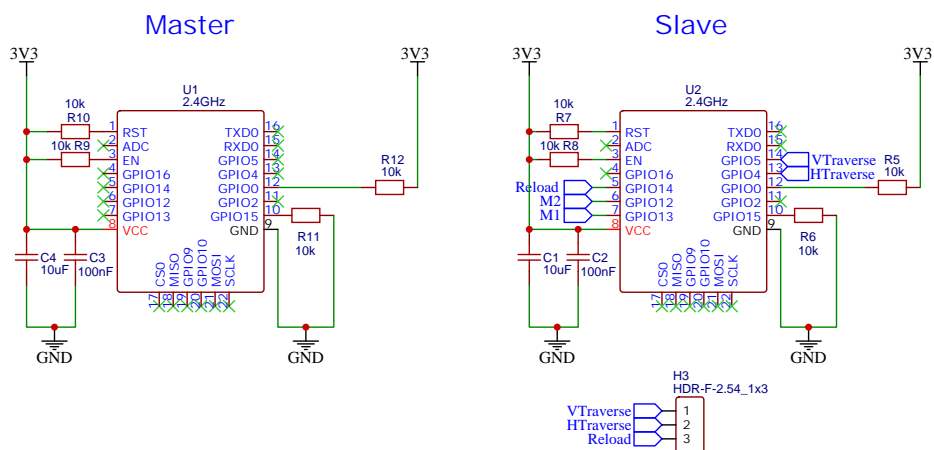
Appendix F Power Schematic and BOM

Table 8 shows the bill of materials (BOM) of the PCB design. Schematic and PCB design is shown in the next two pages.

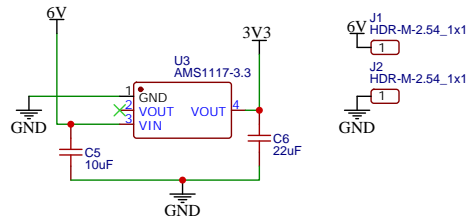
Table 8: Bill of Materials (BOM) for the PCB design.

ID	Name	Designator	Quantity	Manufacturer	Supplier Part	Price (RMB)
1	10 μ F	C1,C4,C5	3	Samsung	C15850	0.066
2	100 nF	C2,C3	2	Samsung	C1711	0.039
3	22 μ F	C6	1	HRE	C6119868	0.25
4	F-2.54_1x3	H3	1	Ckmtw	C146690	0.56
5	F-2.54_1x4	H4,H5	2	ZHOURI	C5116530	0.48
6	M-2.54_1x1	6V,GND	2	BOOMELE	C81276	0.046
7	470 Ω	R1,R4	2	FOJAN	C2909361	0.003
8	150 Ω	R2,R3	2	YAGEO	C114523	0.013
9	10 k Ω	R5-R12	8	YAGEO	C84376	0.012
10	AMS1117-3.3	U3	1	AMS	C6186	1.21
11	2.4GHz	MASTER,SLAVE	2	Ai-Thinker	C82891	29.54

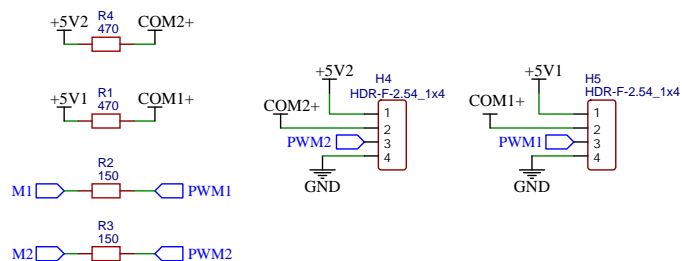
ESP-12F Modules




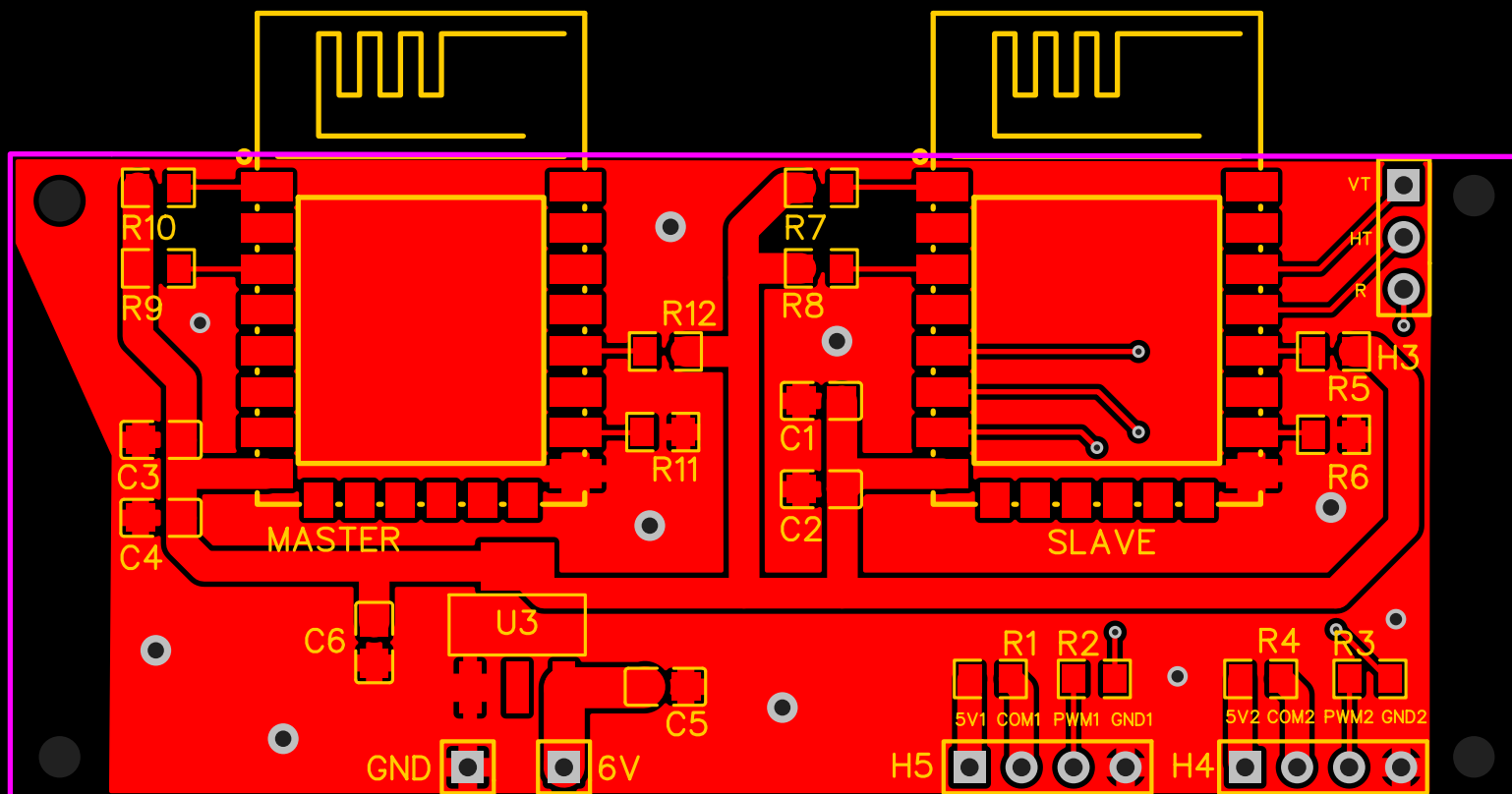
Power Conversion



48 V PWM Control



TITLE: Dodgebot PCB		REV: 1.0
	Company: Team 41	Sheet: 1/1
	Date: 2025-05-16	Drawn By: Loigen Sodian



Appendix G Base and Turret Schematic Design

The complete schematic for the base and turret design is shown in the following pages.

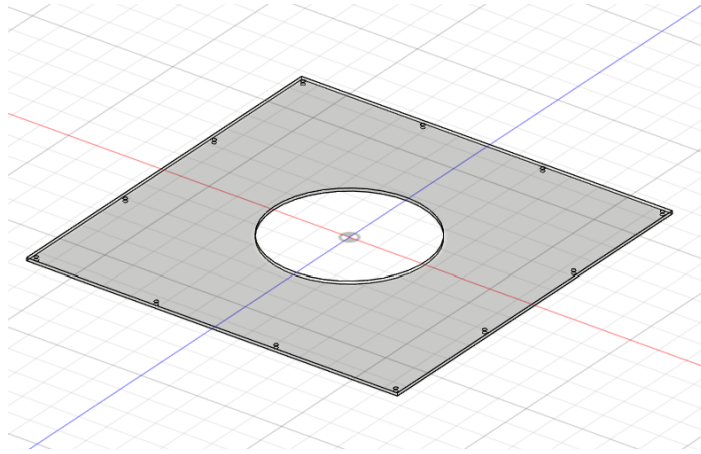


Figure 1. 3D model of the top cover for the base

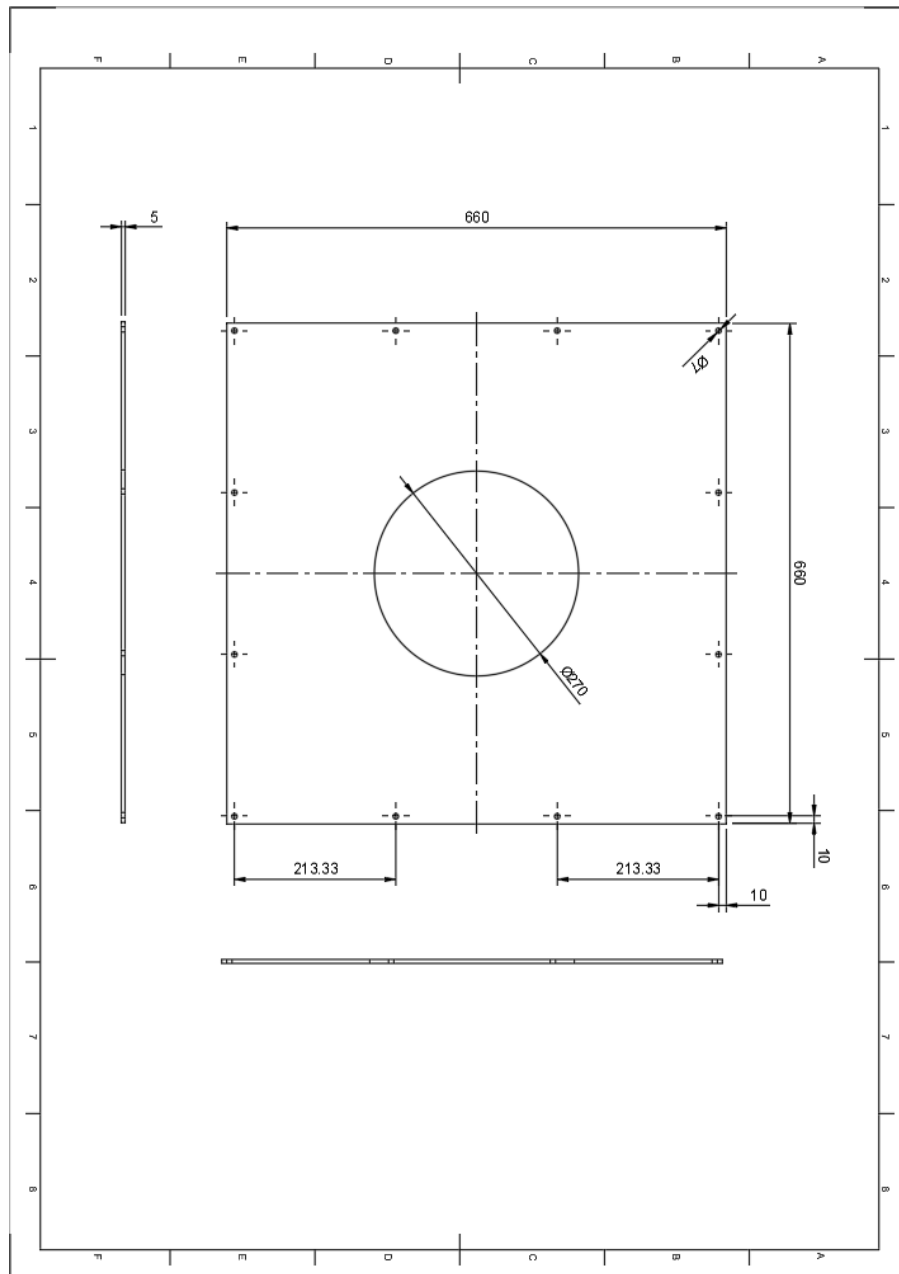


Figure 2. The engineering drawing of the top cover for the base

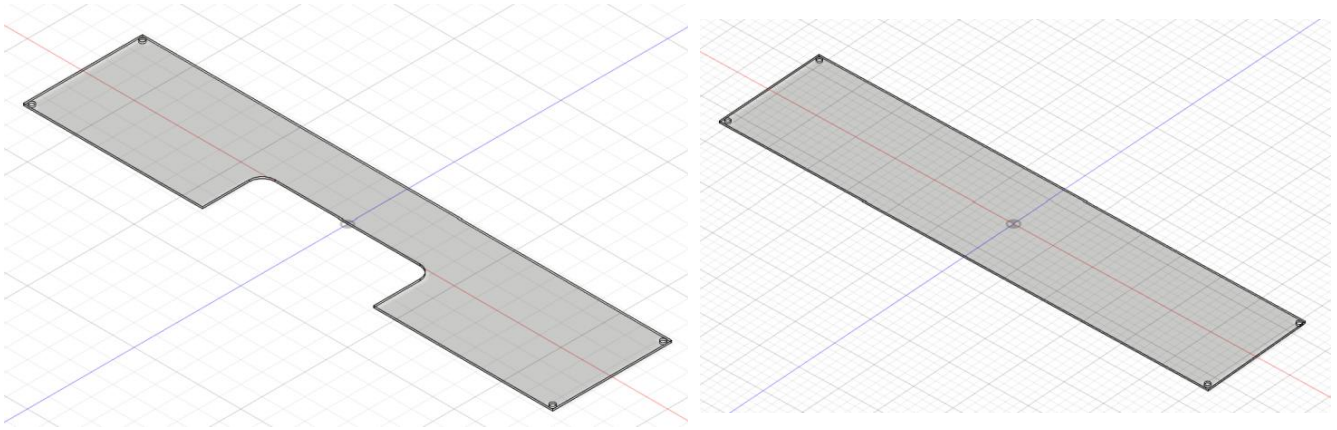


Figure 3. 3D model of the side wall with cable outlet (left) and without outlet (right) for the base.

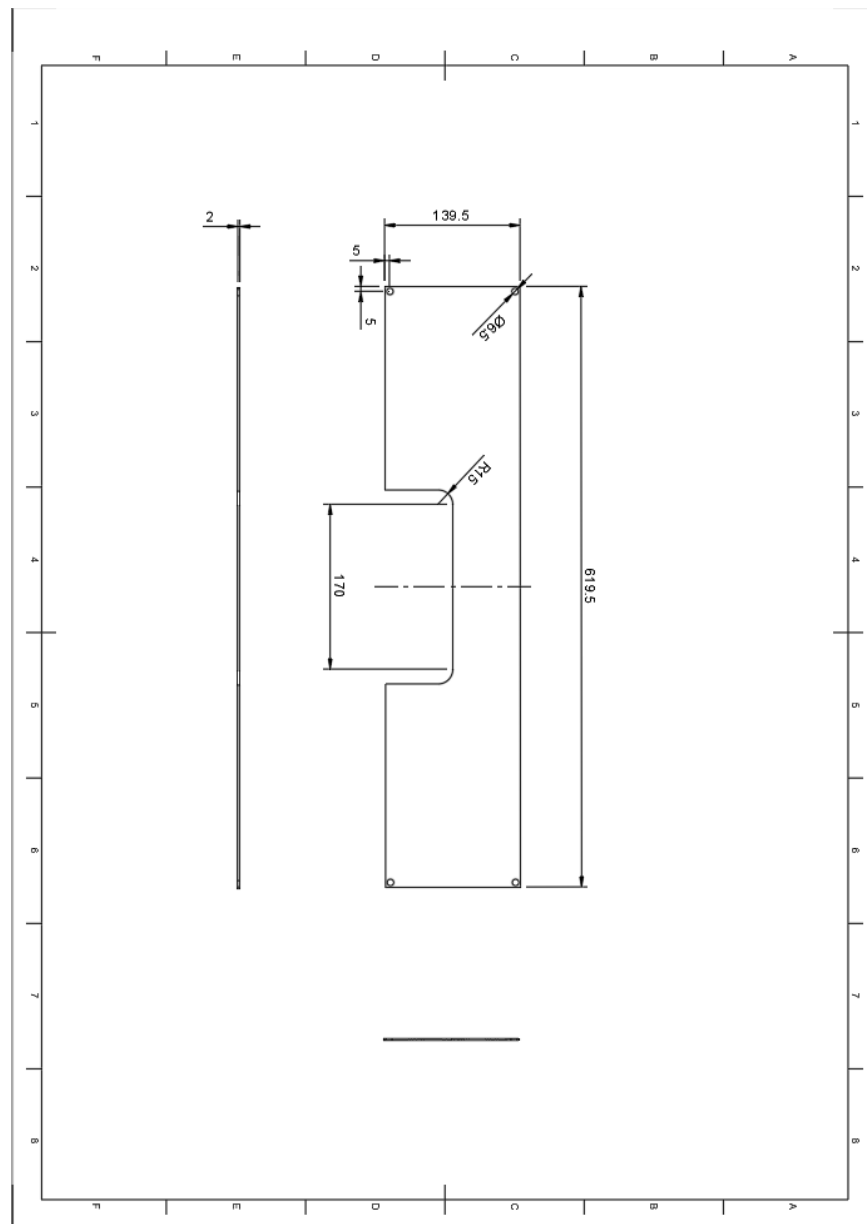


Figure 4. The engineering drawing of the side wall with cable outlet for the base

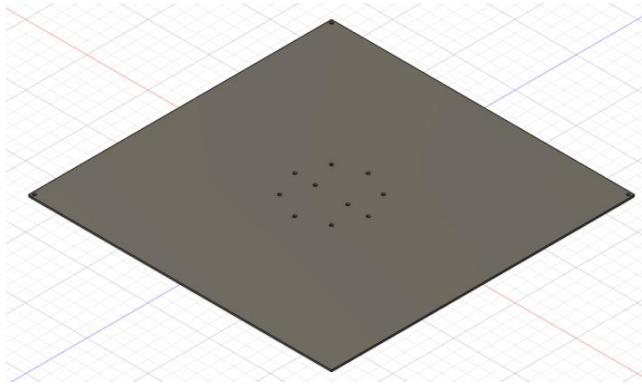


Figure 5. 3D model of the bottom board for the base

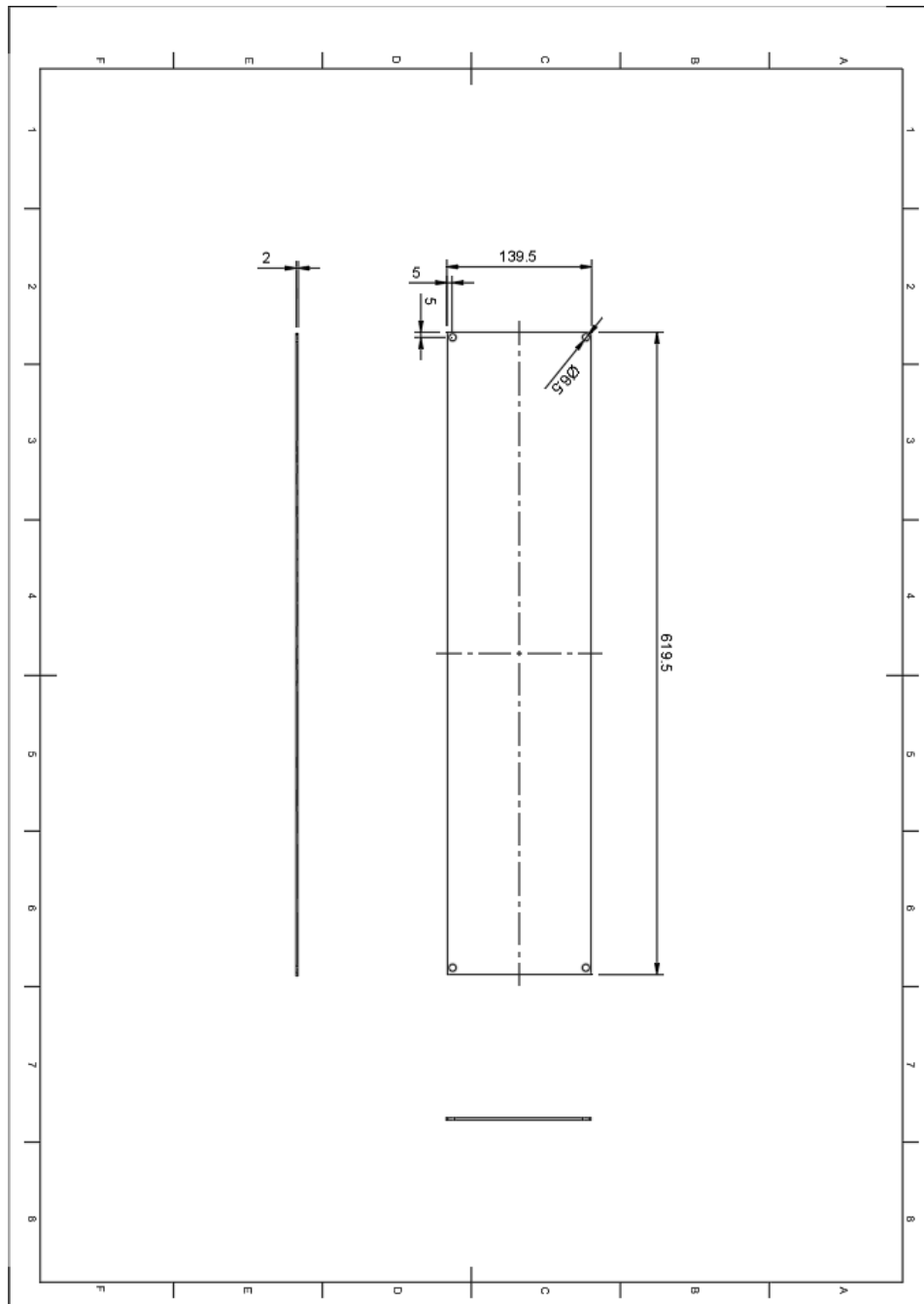


Figure 6. The engineering drawing of the side wall for the base

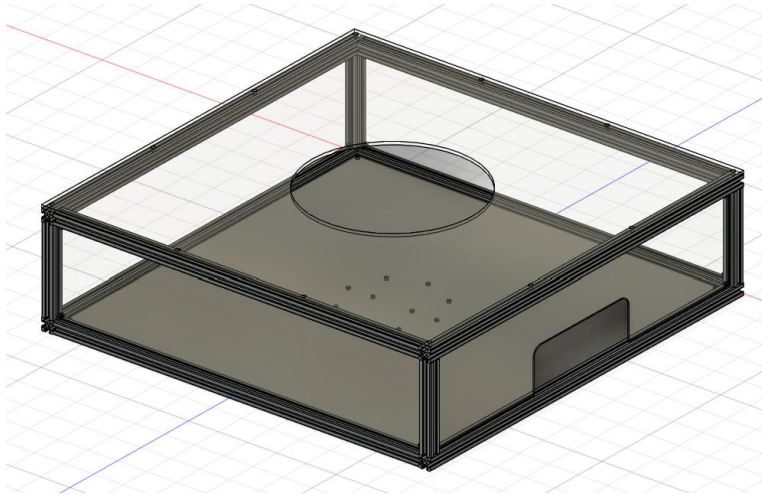


Figure 7. 3D model of the base with aluminum frame

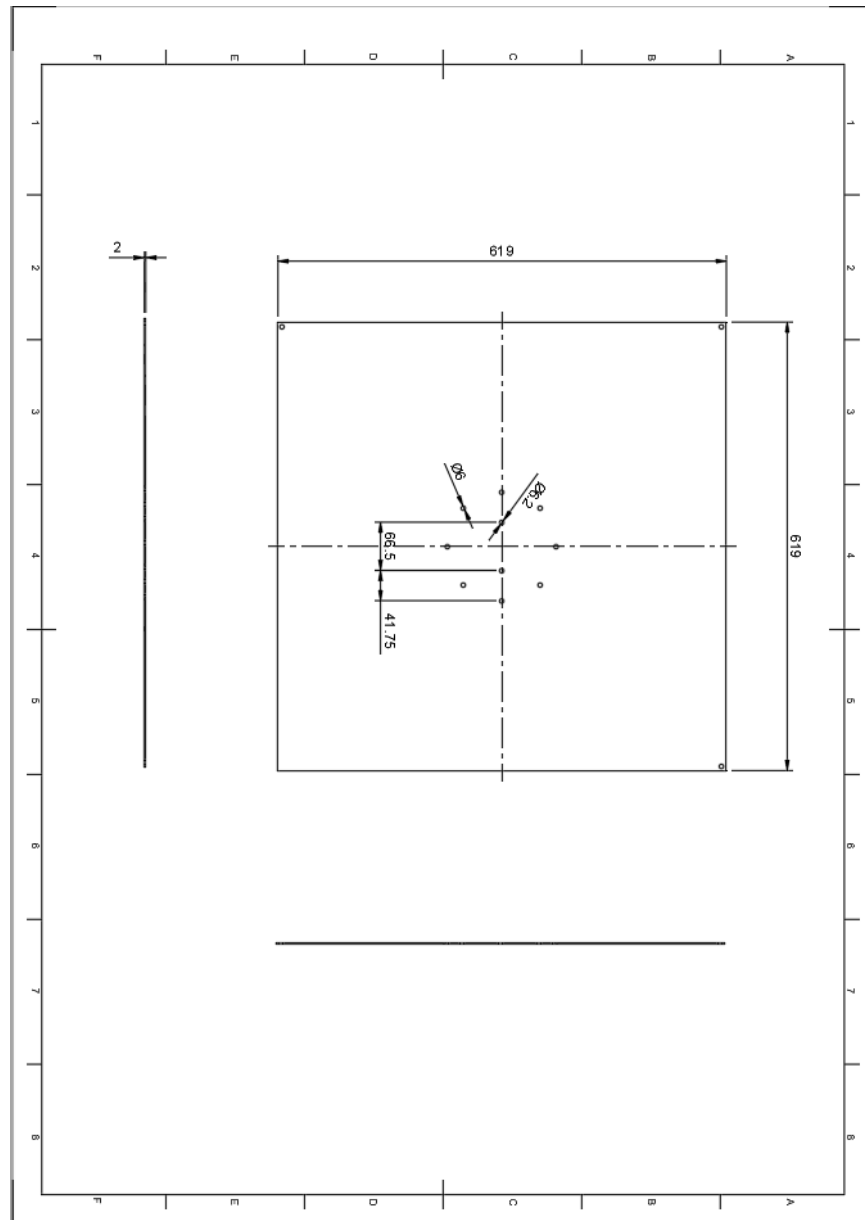


Figure 8. The engineering drawing of the bottom board for the base

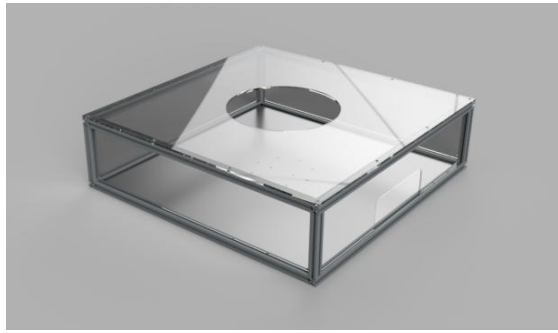


Figure 9. The rendering picture of the base with aluminum frame

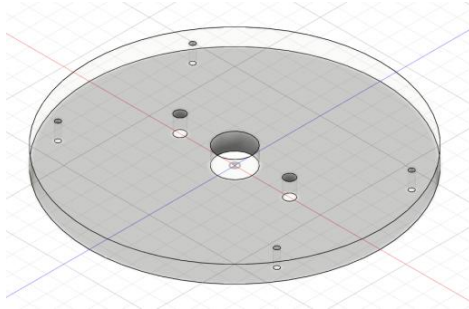


Figure 10. 3D model of the circular base for the gun

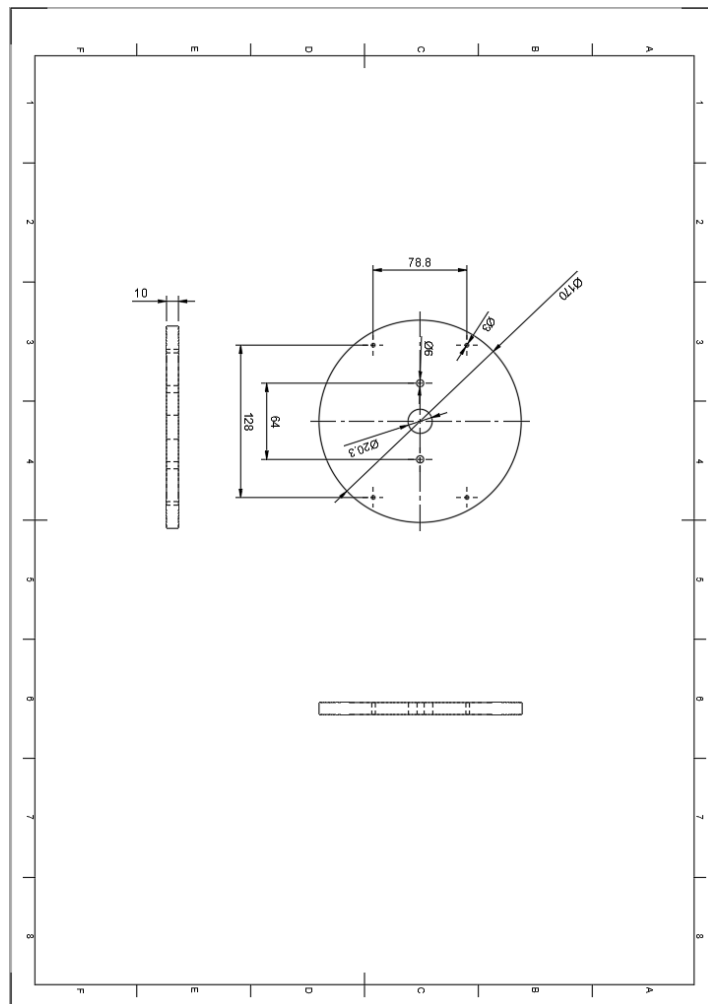


Figure 11. The engineering drawing of the circular base for the gun

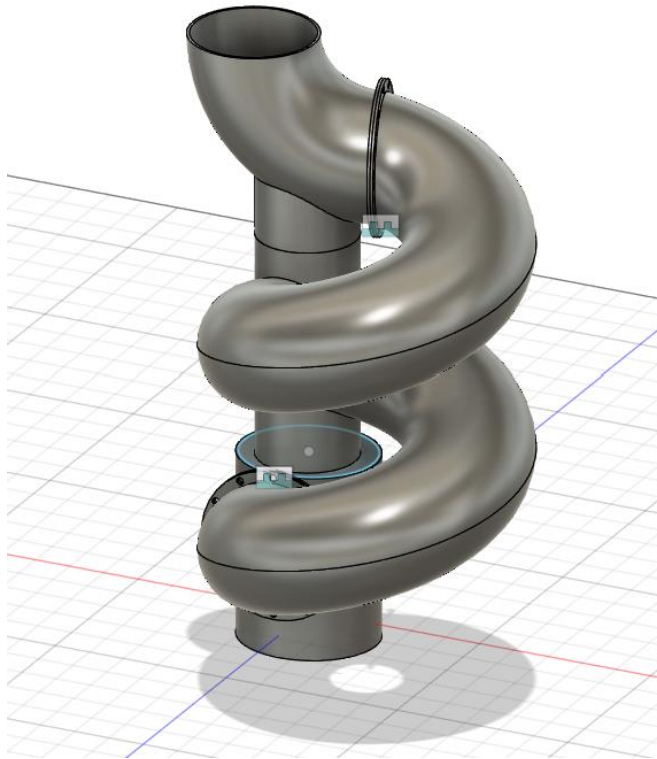


Figure 12. 3D model of the spiral pipe



Figure 13. The rendering picture of the spiral pipe

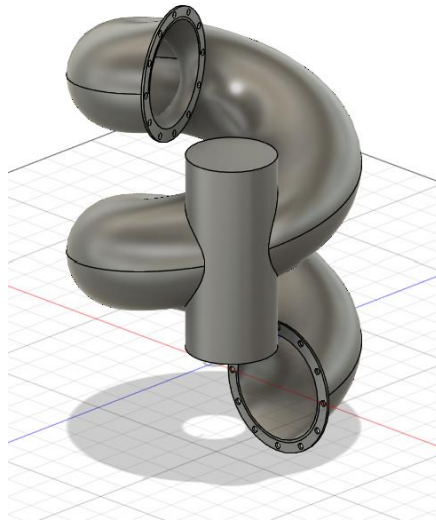


Figure 14. 3D model of the body of the spiral pipe

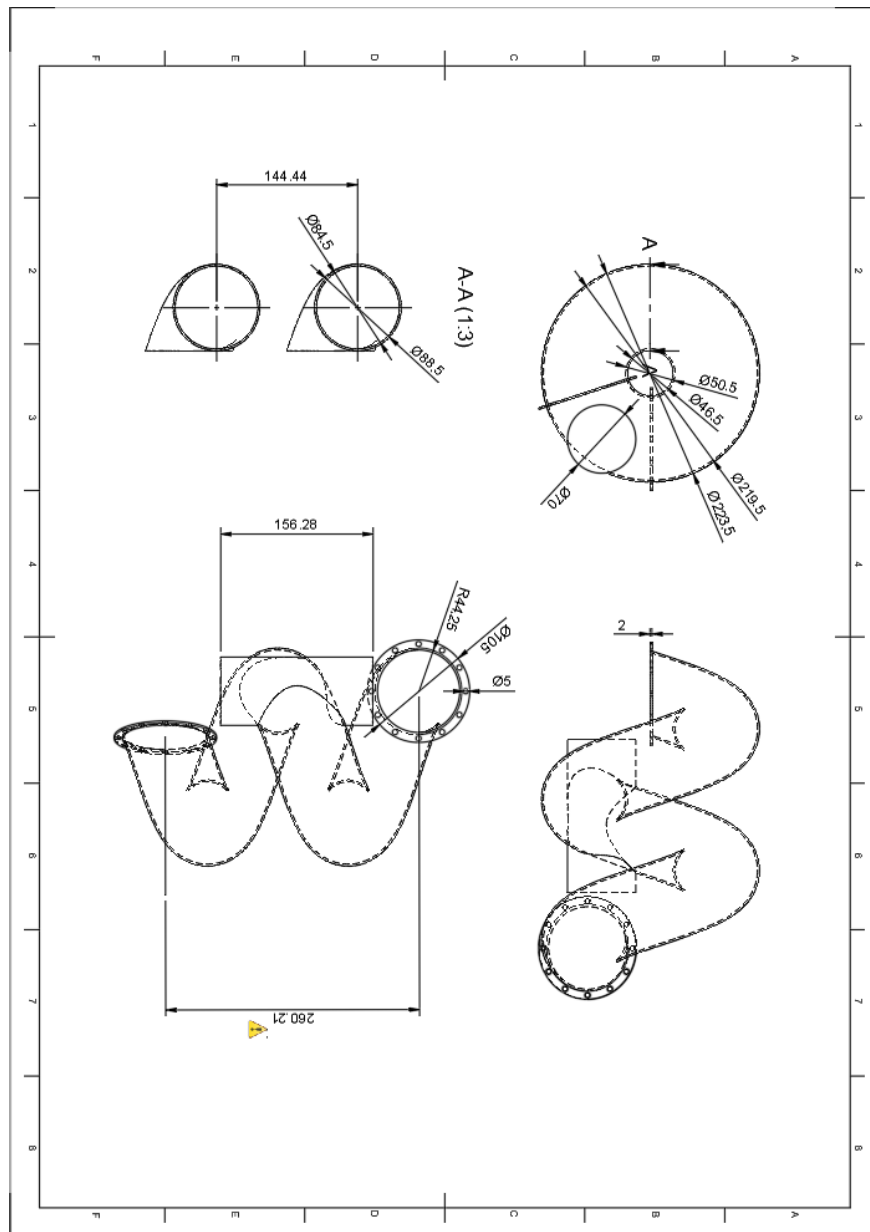


Figure 15. The engineering drawing of the body of the spiral pipe

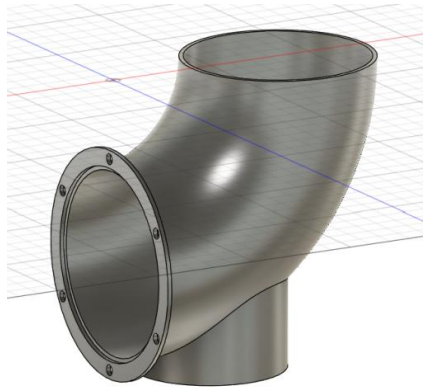


Figure 16. 3D model of the inlet of the spiral pipe

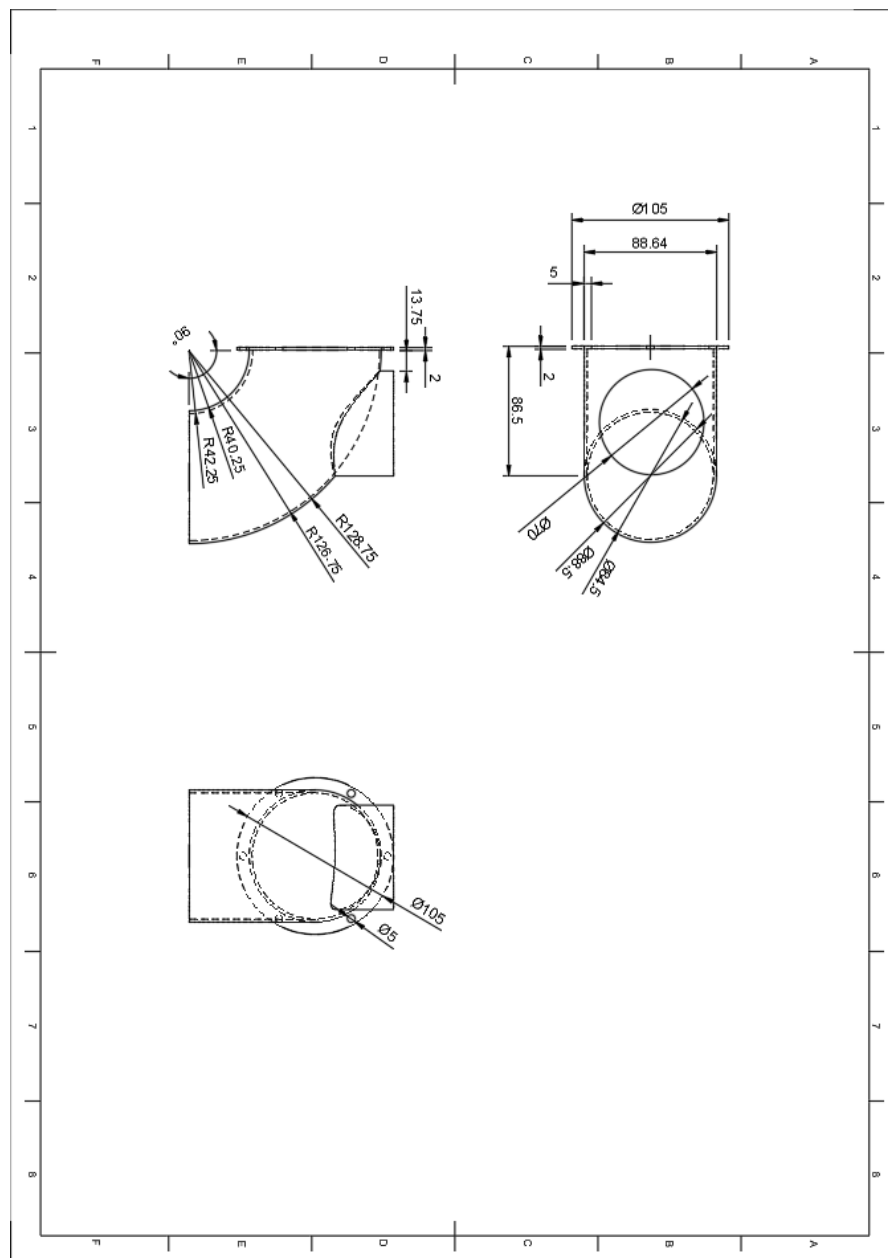


Figure 17. The engineering drawing of the inlet of the spiral pipe

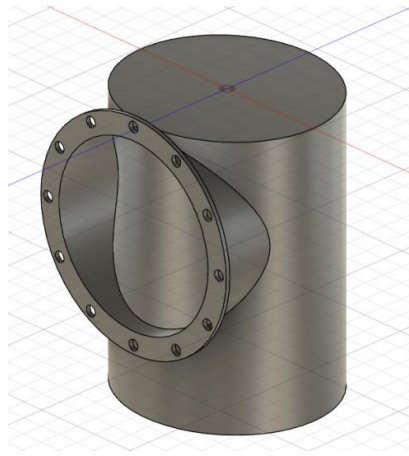


Figure 18. 3D model of the bottom end of the spiral pipe

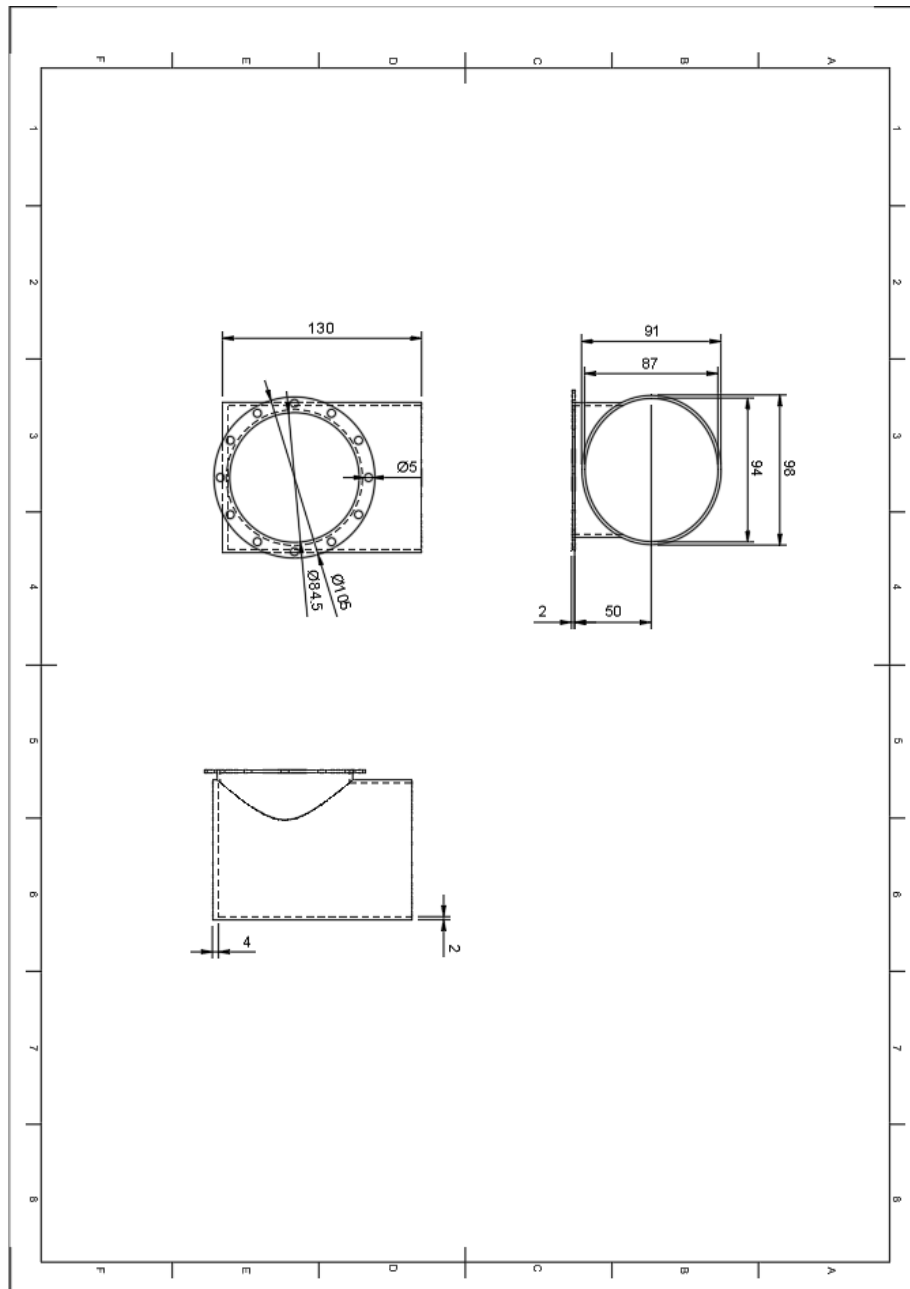


Figure 19. The engineering drawing of the bottom end of the spiral pipe