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

We present the motivation behind developing a dodgeball launcher and outline the high-level requirements used to evaluate the final design. Section 1.3 also describes the structure of this report.

1.1 Problem and Solution Overview

Traditional dodgeball lacks accessibility, requiring at least two players and making solo play impossible. Recent alternatives like AR-based dodgeball [1] and motion tracking systems [2] aim to modernize the game but face limitations, as AR depends on controlled setups, and motion tracking struggles with latency and lighting issues [3].

To address these, we introduce *Dodgebot*, an autonomous robotic platform integrating a thermal camera, motorized turret, and AI control. Using the ESP-12F [4], the system detects targets without wearables. A 360° rotating turret ensures flexibility, while dual motors launch projectiles at high speeds. Powered by a portable 48 V supply, *Dodgebot* offers a safer, more accessible, and scalable approach to robotic dodgeball.

1.1.1 Visual Aid

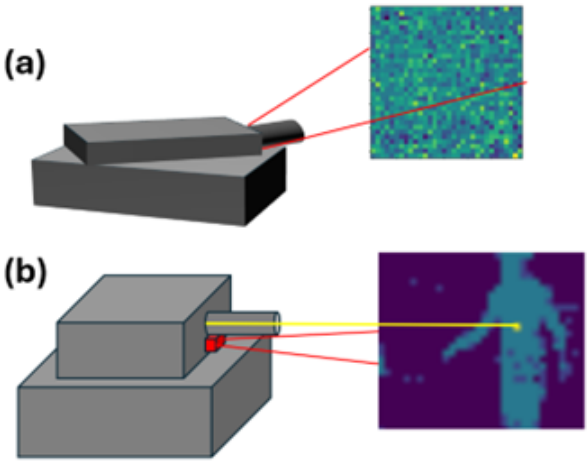


Figure 1: In (a), the robot scans the environment using a thermal camera mounted beneath the gun barrel. If no target is detected, the robot does not fire. Upon detecting a target, it aligns with the target's center of mass and fires immediately.

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 by the ESP-12F MCU) will deem the target non-existent in this context. If so, the robot will rotate to a new firing position. If an object of appreciable size 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).

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 ¿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 and ethical concerns of *dodgebot*. Section 3, Section 4, Section 5, and Section 6 discusses each component: 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.

2 High-level Design

2.1 Block Diagram of *Dodgebot*

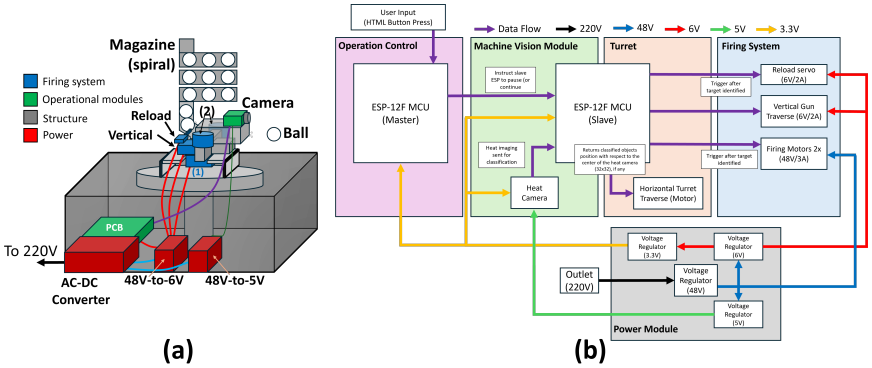


Figure 2: (a) the 3D schematic of the robot. (1) is the horizontal traverse servos and (2) shows the firing motors. The "PCB" block houses the ESP-12F and the 6 V to 3.3 V converter. Colored traces follow the legend in (b).

The 3D illustration and block diagram are shown in Fig.2a and Fig.2b, respectively. The robot launches a tennis ball (diameter 63.5 mm) using two DC motors. When the slave ESP detects targets, appropriate control signals are sent to initiate firing. Based on their functions in Fig.2a, components are grouped into five modules: *operation control*, *machine vision module*, *turret*, *firing system*, and *power module*, as shown in Fig.2b.

2.2 Operation Control Module

This module determines whether the slave ESP should activate its target detection and firing routines. It creates an access point (AP) that users can connect to for control. Through an HTML interface, users can toggle the robot's state between active and inactive via a button on their mobile device.

2.3 Machine Vision Module

This module uses a thermal camera and an ESP-12F for image classification. Mounted on the gun barrel, the camera shares the gun's line of sight. At each viewpoint, the ESP detects objects with heat signatures significantly above ambient levels by analyzing pixel-to-pixel temperature differences. If a target is found, the slave ESP issues commands to align the turret horizontally and vertically, then reload and fire.

2.4 Turret Module

The turret module controls gun rotation and firing. The ESP sends commands to the horizontal and vertical traverse motors to center the target. If no target is found, the

turret rotates to a new angle for scanning. When a target is detected, the ESP activates the reload motor to load one tennis ball into the chamber. The ball then contacts the counter-rotating firing motors, launching it outward. The system continues tracking and firing until the target exits the field of view. Afterward, the ESP resets the elevation, switches off the firing motor, and resumes scanning.

2.5 Firing System Module

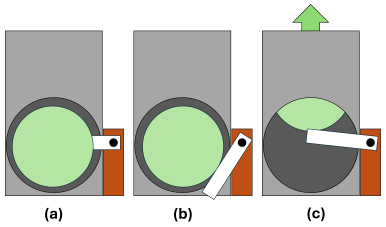


Figure 3: Reloading process of the robot. The chamber is initially empty, and the ball rests on the reload servo's spoke (a). When reloading, the spoke turn backwards to let the ball into the chamber (b). The loaded ball is launched into the firing motors in (c).

The firing system includes the ball reload system, the firing mechanism, and ball storage (magazine). Reloading is controlled by an MG996R servo with extended spokes, ensuring only one ball enters the chamber at a time. Fig. 3 shows three stages: initial position (a), chamber loading (b), and feeding the ball to the firing motors (c). The firing mechanism uses two horizontal motors rotating in opposite directions to launch the ball at high-speed. The elevation is controlled by a single servo, capable of lifting 40 kg/cm, attached to the gun and reload tube.

2.6 Power Module

This module supplies power to the motors and ESP MCUs from a wall outlet via voltage regulators. An AC-DC converter converts AC to 48 V DC. Part of this powers the firing motors directly, while the rest is stepped down gradually: from 48 V to 6 V or 5 V. The reload servo and traverse servos use 6 V, the thermal camera uses 5 V USB, and the two ESPs are powered at 3.3 V via an AMS1117-3.3 regulator stepping down from 6 V.

2.7 Ethics

Dodgebot addresses safety and ethical concerns related to projectile impact, turret movement, and user privacy. It uses soft tennis balls and limits projectile speeds, following the IEEE Ethics Guidelines [5] and ASTM standards [6] to minimize injuries. The robot boasts internal wiring to prevent electrical hazards, complying with OSHA guidelines [7]. Privacy is ensured by processing thermal images in real-time without storing data, requiring user consent before tracking, aligning with ACM Ethics Guidelines [8]. By adhering to industry standards, legal regulations, and local policies, Dodgebot remains a safe, ethical, and responsible system.

3 Operation Control Module

Note that this module is not covered in the initial product proposal. The implementation was first suggested by Prof. Timothy Haw-yu Lee during the initial product 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. Therefore, we implemented the panic button wirelessly. We use 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 [9]. 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 [10].

3.2 Panic-pause Feature Description

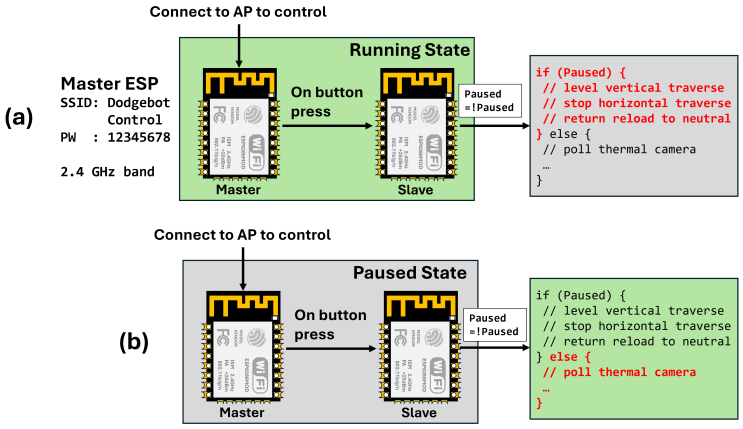


Figure 4: When the robot is running (a), the user can connect to the specified SSID and password to open the HTML interface. When the user press the "TOGGLE PAUSE" button, the master ESP transmits data to the slave ESP, and the slave ESP will stop all servos, motors, and image polling. If the user wishes to continue, they may press the button again, which will toggle the slave ESP to continue as normal.

The panic pause feature is implemented by allowing the master ESP to be the AP emitter, and the diagram is shown in Fig. **??**. 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 (Fig. 3a), the horizontal

traverse servo is stopped, and the firing motors stop spinning. 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.

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. Once connected, each ESP is assigned a unique IP, allowing the user to flash them via the Arduino IDE. The OTA setup code is shown in Appendix D. This process is illustrated in Fig. 5.

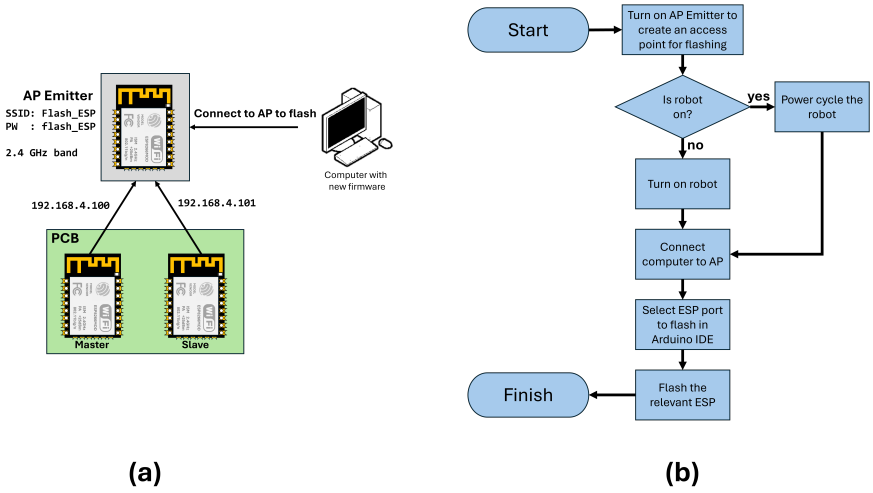


Figure 5: The OTA flashing mechanism. In (a), the two ESPs in the robot connects to an AP Emitter with a set of SSID and password. The computer (which contains the new firmware) connects to this AP. To flash either of the two ESPs, the user can select the appropriate ESP in the Arduino IDE and initiate flash accordingly. (b) shows the flowchart of the OTA flashing process.

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 mobile device, and the interface is made user-friendly. The OTA flashing feature allows the delivery of new firmware updates without having to introduce 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

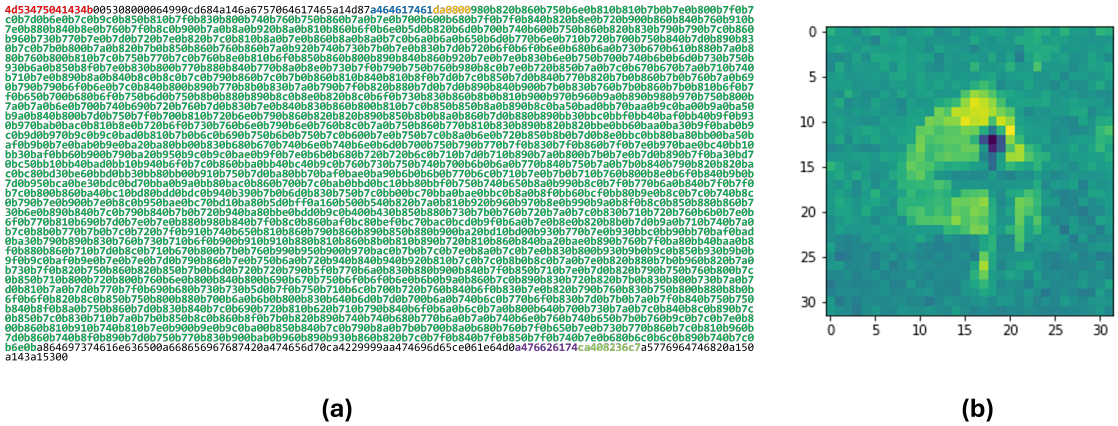
We now introduce the "brain" of the robot, responsible for controlling the various servos/motors while also being the main processing power behind the object detection algorithm.

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 this section.

4.1.1 Image Polling Through Websocket

The thermal camera (RT-Thermal) streams real-time image data over Wi-Fi, which is why we use the ESP-12F. The data is encoded in the MessagePack format [11] and transmitted via the WebSocket protocol (RFC6455) [12]. To interface the ESP with the thermal camera, the ESP connects to the camera's access point (AP) and uses Arduino's built-in `ArduinoWebsockets` library to receive WebSocket data. Fig. 6a shows a typical WebSocket payload from the camera. The image is reconstructed by converting the binary data (highlighted in dark green) into an array of `uint16_t` values with a total length of 1024. This array is then visualized as a heatmap in Fig. 6b. The 1D array is reshaped into a bijective 32-by-32 2D array, where each element $[i, j]$ corresponds to the $[i, j]$-th pixel in the thermal image. This 2D array forms the input for our object detection algorithm.



(a) (b)

Figure 6: An example of the streamed raw data (a). Red binaries are ASCII for `MSGPACK`, blue is ASCII for `data`, yellow is 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.

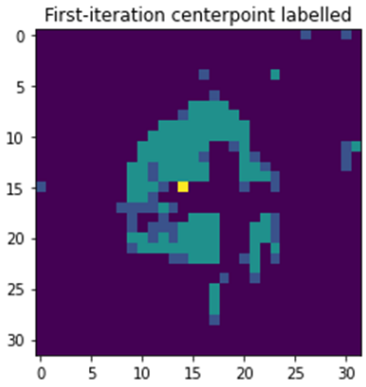


Figure 7: The trinary image along with the first iteration centroid $[i_{\text{mean},1}, j_{\text{mean},1}]$ labelled (yellow).

4.1.2 Centroid Detection Algorithm

We first filter images that do not contain blobs (e.g., background 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 assume the recorded temperature is fairly uniform across all pixels, hence no target is present. If the image exceeds this limit, an anomaly is detected, and it will then undergo *trinary mapping*. The image array (comprised of raw values from 2000 to 3000) is transformed into a trinary image according to (1).

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

Then, we calculate the mean value of each rows and columns. A higher mean value indicates a higher presence of the blob in the trinary image. We select 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. 7.

Unfortunately, the predicted centroid may deviate due to contributions from extraneous blobs. We filter these points from the final centroid calculation 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 $\langle d_r \rangle$ and $\langle d_c \rangle$. We will use this value to filter out rows or columns that lie further than these values respectively. After filtering, we now calculate again the predicted centroid to get $[i_{\text{mean},2}, j_{\text{mean},2}]$, i.e., the final centroid (see Fig. 8). This is an improvement over Fig. 7 as now the predicted centroid lies within the blob region. We also determine the offset $[i_{\text{mean},2} - 16,\ j_{\text{mean},2} - 16]$ to find the centroid's deviation from the image center. This value is then processed by the ESP into values meaningful for the servos and motors.

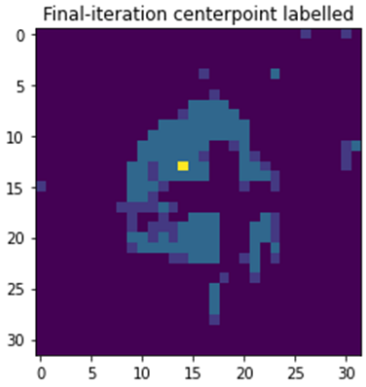


Figure 8: The trinary image along with the second iteration centroid $[i_{\text{mean},2}, j_{\text{mean},2}]$ labelled (yellow). The centroid now lies within the blob.

4.2 Motor and Servo Control

The pixel offset alone is insufficient to determine the required rotation angle for the gun. Since the servos are controlled via PWM, we must translate pixel displacement into corresponding PWM instructions. We found that an offset of one pixel corresponds to approximately $1.5°$. Therefore, a shift of n pixels equates to a rotation of $n \times 1.5°$. The servos used—RDS3225 for vertical and horizontal traverse, and MG996R for the reload mechanism—accept PWM pulse widths in the range of 1000 to 2000 μs [13], [14]. Using this relationship, we constructed a reference table (Table 1) that maps pixel offsets to corresponding PWM pulse widths in microseconds.

Table 1: Tabulation of PWM widths (in μs) for (a) horizontal traverse and (b) vertical traverse and reload servo. A separate entry is required for the horizontal traverse servo as it is continuously-rotating.

(a)

Servos	Stop	CCW	CW
Horizontal	1500	1800 ($6°$/s)	1200 ($10°$/s)

(b)

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

We use Arduino's `Servo` module to write PWM signals. If the centroid lies within two pixels of the center, the gun remains stationary. Otherwise, we implement a proportional (P)-controller combined with velocity estimation to determine servo movement. The velocity $v_{i,j}$ is calculated as the displacement between the last and current pixel positions.

We then predict the blob's position 50 milliseconds ahead. Applying proportional gains $K_H = 15$ and $K_V = 9$ for horizontal and vertical traverse respectively, the control signals are computed as $K_{H,V} \times \text{pred}_{i,j}$. Using the tabulated values in Table 1, we constrain these values within ± 300 for horizontal and ± 200 for vertical traverse, giving `hspeed` and `vspeed` respectively. This P-controller produces smoother turret rotation compared to our previous iteration which suffers from discrete jumps between maximum speed and stationary. Additionally, velocity prediction allows us to lead the aim ahead of the moving target. Note that firing motors are controlled by a different PWM mechanism, which will be detailed in Section 5.

4.3 Design Verification

Requirement No. 1 is trivially satisfied, as the code automatically rejects image with deviation less than 90. For Requirement No. 2, we measured the F1-confidence score by sampling ten image sets, five containing valid targets and five with only background noise at distances of 5, 6, and 7 meters. The confidence scores were 100% at 5 meters, approximately 88% at 6 meters, and dropped to 57% at 7 meters (see Table 2). We extended Requirement No. 3 to include the full processing pipeline—from polling the image to relaying PWM signals to the servos—to demonstrate system speed. The entire process takes about 30 ms, which is significantly faster than the original requirement. Note that requirement No.4 is not implemented in the design, since we implemented velocity calculation, which is able to catch fast targets at 5 meters. Our velocity prediction can compensate fast moving targets, even with the relatively low camera resolution.

Table 2: F1-score along with true positives, true negatives, false positives, and false negatives data for three distance measurements.

Distance	True Positives	True Negatives	False Positives	False Negatives	F1-score
5 meters	5	5	0	0	100%
6 meters	4	5	1	0	88%
7 meters	2	5	3	0	57%

4.4 Conclusion

We detailed how the ESP collects data from the thermal camera, identifies the centroid of detected blobs (if any), and controls the servos and motors using a combination of P-control and target velocity estimation. Our algorithm has been successfully verified under various conditions. Notably, the entire algorithm fits within the 32 KiB instruction memory of the ESP-12F, leaving sufficient space for OTA updates and additional smaller programs. This compactness offers great flexibility by eliminating the need for an external processor to compute the centroid, while also keeping costs low—an ESP-12F MCU costs under $2.

5 Firing System Module

The main component of *dodgebot* is its ability to fire multiple rounds at high-speed. This section discusses the firing system module, from its ability to fire at long-range targets to its innovative spiral magazine design to store extra tennis balls. Section 5.1 explains how the firing motors can be programmed by the ESP to either rotate or stop depending on the PWM signal. Section 5.2 discusses the detailed 3D structure of the firing system i.e., the gun and the spiral magazine design. Section 5.3 verifies that our design conforms with the requirements and verification table from Appendix B. Finally, Section 5.4 summarizes this section.

5.1 Controlling the Firing Motors via PWM

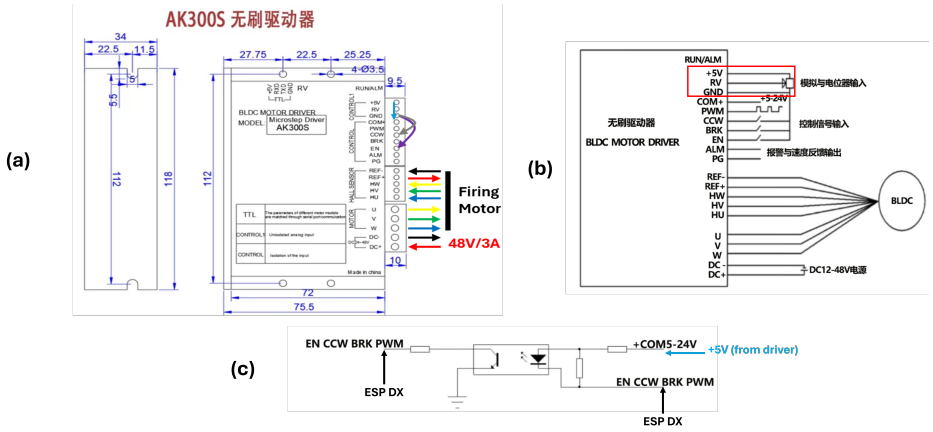


Figure 9: (a) AK300S driver schematic. To rotate the motor counter-clockwise, the GND to CCW connection must be made (gray arrow). To control the driver using PWM, COM+ must be connected to any +5 V source (light blue). (b) example circuit, red box shows how a potentiometer can be used to control the driver. (c) implementation of the PWM control by ESP. DX refers to any of the digital pins of the ESP. Image source [15].

Each firing motor is driven by an AK300S BLDC driver [15], controlled via PWM from the ESP-12F. During the robot's development, we encounter three distinct development phases. First, we tried using an MCP41010 digital potentiometer [16] to directly control the driver. This results in a simple circuit, but back-EMF (which may cause dangerous voltage spikes) often occurs if the motor is not slowed-down in small increments. Furthermore, the MCP41010 requires three digital pins from the ESP (which already has limited pins) to operate, and is very expensive (14 RMB each). So, we switched to PWM control using the driver's built-in PWM pin, reducing complexity and cost. However, back-EMF still occurs when fully stopping the motor, even if the changes are incremental. Hence, we implemented a combination of PWM and BRK control. By connecting the ESP to both the PWM and BRK pins of the BLDC driver. Since BRK is active low, the ESP either pulls this pin to the ground or leave the pin floating to stop or run the motor respectively. This

allows us to safely modulate speed and stop the motor completely, solving the back-EMF issues at the cost of one extra pin. The design iteration is shown in Fig. 10.

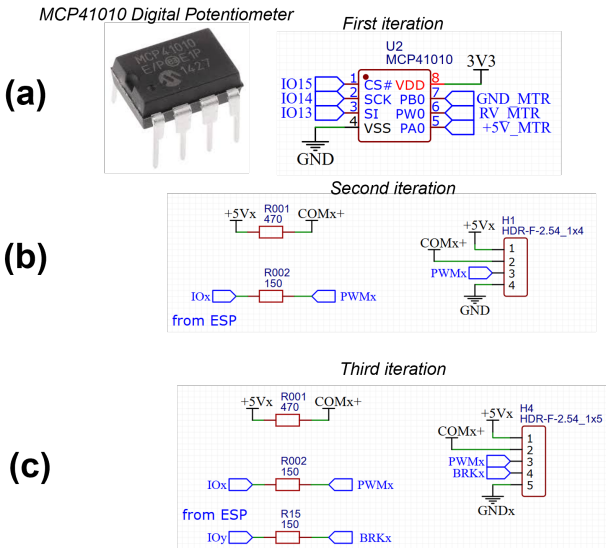


Figure 10: (a) the first iteration, MCP 41010 digital potentiometer. (b) second iteration, PWM control. (c) Third iteration, PWM and BRK control.

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

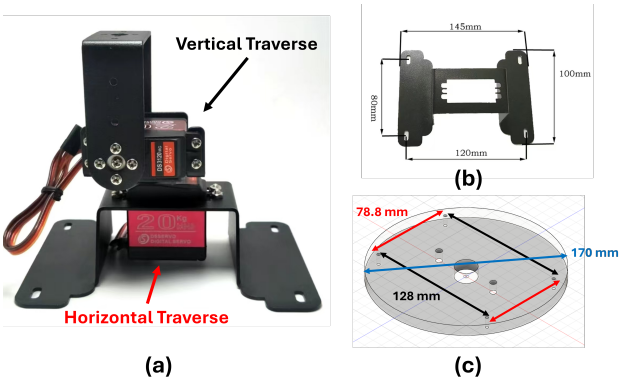


Figure 11: The gun platform (a) and the dimensions of the supporting leg (b). (c) shows the acrylic mount used to hold the platform in (a) on the robot's base. Detailed schematics are supplied in Appendix G.

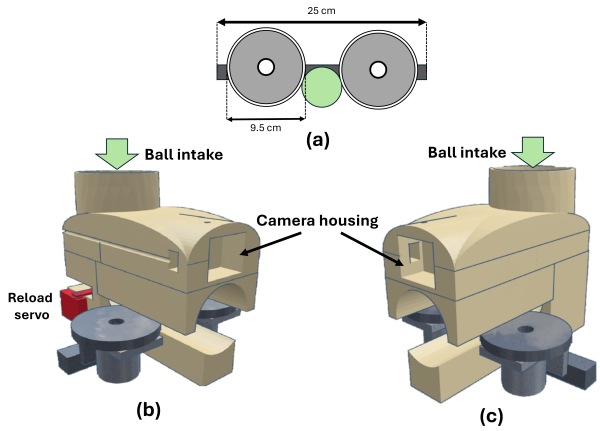


Figure 12: The gun dimensions (a), along with its housing, shown as a 3D render in (b) and (c), each showing the right and left side of the whole gun structure accordingly. For (b) and (c), components colored in gray is the gun while components in yellow is the gun chamber. Additionally, (b) also shows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

.

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. Initially, we planned to house the gun on a completely-closed turret, as shown in Fig. 16 (Appendix G) in our first iteration. However, due to time constraints and aesthetic concerns, we finally opted for a more exposed design where the whole gun is not covered by any form of enclosure, as shown previously in Fig. 11.

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 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 first experimented with an open-top semicircle design (see Fig. 18 in Appendix G), where it functions sim-

ilarly 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 also tried implementing 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 (see Fig. 17 in Appendix G). Therefore, we employ a spiral shape for our magazine design in this third iteration. 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.

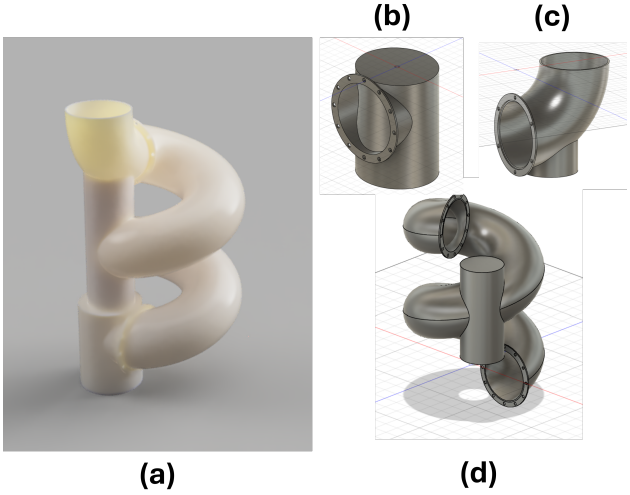


Figure 13: The spiral 3D render (a), which is comprised of three parts. (b) is the bottom connector that connects to the gun chamber, (c) is the top connector that allows users to replenish the magazine, and (d) is the spiral that acts as the tennis ball magazine. Detailed schematics are supplied in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

5.3 Design Verification

Using the ESP module, we programmed the vertical traverse to hold the turret at a 20° incline for 20 seconds, satisfying Requirement No. 1 from Appendix B (gun rotational adjustment). The servo maintained this position while drawing only 0.17 A, well below its maximum rated current. Our tabulated values were also verified. As shown in Table 1, sending PWM signals of 1500 ± 25 μs resulted in a 10° change in vertical angle (Requirement No. 2), and signals of 1500 ± 300 μs produced horizontal rotation at $10^\circ/\text{s}$ (Requirement No. 5). Requirement No. 3 was confirmed by measuring the time needed to level the gun from a -20° decline while the firing motors were active. Averaging five trials, the servo completed the adjustment in just 64 ms, demonstrating the gun's capability to track targets vertically at high speed.

We also tested if the servo is able to spin 360 degrees in 20 seconds (requirement No. 4). This can be achieved by setting PWM signals 1500 ± 500, which we observe may rotate the

turret within 17.2 seconds. While doing this we found an issue where the wiring would get entangled whenever the turret rotates. This is solved by implementing a code where the turret would be spun the opposite direction of its rotations to detangle the wires. In the final product, we scaled the rotational speed down to improve camera tracking. Finally, we choose to not include requirement No.6 since our velocity calculation (for object prediction) is enough to track fast-moving targets, hence we may not suffer from losing target constantly.

We also tested the projectile reload mechanism as specified in Requirement No. 1. Using the Arduino IDE, the console prints "object detected" when the ESP detects an object. We loaded 10 image samples containing detectable signatures, and observed that the reload motor triggered 100% of the time upon successful detection. We also verified the reload servo's operation in line with Requirement No. 2: when aimed at a valid target, the gun fires one ball every 1.5 seconds—faster than stated in Appendix B, yet consistently maintained. The reload servo is strong enough to handle a stack of 10 tennis balls, and we observed no misfires during testing, fulfilling Requirement No. 3. This was tested by positioning a person 5 meters from the robot, measuring the reload interval, and confirming that only one ball was loaded per cycle. Finally, we verified Requirement No. 4: the robot can consistently hit targets within 5 meters, with a measured ball velocity of 10 m/s. This performance is guaranteed when mounted on a 1-meter platform, and the robot still achieves an 80% hit rate at ground level, thanks to its strong firing power.

We chose not to implement Requirement No. 5 i.e., to stop the firing motor when no target is detected. In high-activity environments, targets are rarely out of view for long, and frequent motor cycling during testing caused wear on the motors and drivers. Repeated on/off switching also risks damaging the power supply due to back-EMF and current spikes. Our reload motor reliably prevents stray balls from entering the chamber, making an automatic firing trigger unnecessary.

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.

We went through two iterations for the gun platform, the first is a full turret enclosure, which is replaced by the more simple gun platform to reduce size. We also went through two iterations for the magazine system. We started with an acrylic L-tube and bowl which functions as the reload tube and magazine respectively. We then revised our design and went ahead with the spiral magazine system for both functionality and aesthetic purposes.

6 Power Module

This section introduces how the power distribution 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

Table 3: 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**

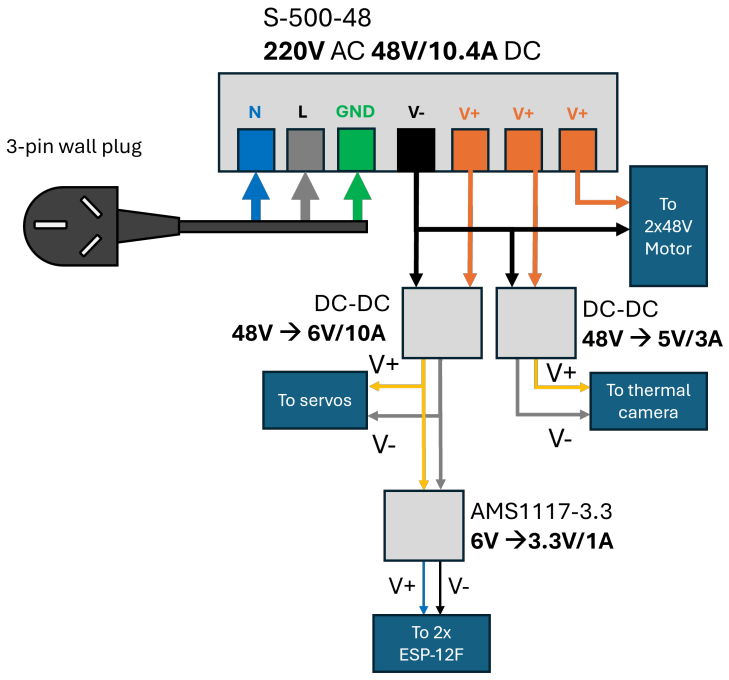


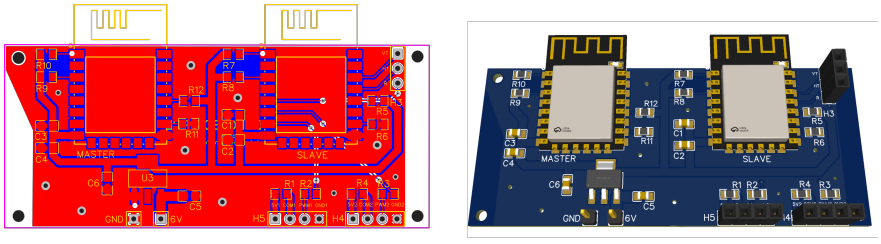
Figure 14: The power distribution diagram. The S-500-48 has 3 V- pins, represented as one V- pin for clarity. Gray blocks indicate converters, while dark blue blocks indicate load components.

Power distribution begins with an AC to 48 V DC converter (S-500-48), which can supply up to 10.4 A. This converter directly powers two 48 V motors, each rated at 2.6 A. A 48 V to 6 V converter is used to power the servos, and a separate 48 V to 5 V USB converter charges the thermal camera. Additionally, two ESP-12F modules are powered via 3.3 V, provided by an AMS1117-3.3 step-down converter that converts from 6 V to 3.3 V. Table 3 lists the maximum current ratings for each module. Actual current draw during normal operation may be significantly lower: each 48 V motor typically draws about 300 mA, combined servo current rarely exceeds 1 A, and the thermal camera has an internal battery. The power distribution is illustrated in Fig. 14.

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 [17], while the latter requires a $10\ k\Omega$ pull-down resistor in IO15, and several $10\ k\Omega$ pull-up resistors for normal operation. 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* [18].

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 naming follows the AK300S pinouts. The "+5V" to "COM+" connections have additional $470\ \Omega$ load resistor, since the output current of the "+5V" pin is about 20 mA [15]. If we assume that the optocoupler LED (see Fig. 9c) has forward voltage 1.2 V, then we obtain the voltage across a resistor R (connecting "+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. 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.



(a) (b)

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 the RDS3225, 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. The actual current draw and voltage drop is shown in Table 4. Both firing motors only draw 0.6 A, which is much less than what is shown in Table 3. This is reasonable since the firing motor spins freely most of the time, and the only resistance it encounters is when firing the tennis balls. Similarly, most components operate at barely half of the maximum current, which shows that the robot's power draw is within safe limits, and hence satisfies requirements No.1-4.

Table 4: Actual current and voltage readings from the device's normal operation.

Component	Voltage	Current Draw	Power Estimate
2× 48 V motors	48 V	0.6 A	28.8 W
2× RDS3225	6.2 V	0.63 A	1.86 W
MG996R	6.2 V	0.25 A	0.62 W
Thermal Camera	5.1 V	0.4 A	2.04 W
2× ESP-12F	3.2 V	0.15 A	0.495 W
Total		**2.03 A**	**33.815 W**

The AMS1117 is capable of converting 6 V, but from Table 4, the output floats around 3.2 V. Nonetheless, 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) [19]. 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, and verified it according to the requirements set.

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* [20], Table 5 shows the labor cost of each members in the group assuming an hourly wage of $20, calculated from February 1, 2025 to May 25, 2025.

Table 5: Labor cost for each member in dollars. The compensation is calculated using the formula outlined in [20].

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

8 Conclusion

We successfully built a dodgeball machine capable of throwing tennis-sized balls at a minimum distance of 5 meters. We acheived 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. Although our dodgeball machine doesnt work in the final demo, we believe that we can make it work if we have enough time to replace the motor that is burnt due to grounding problems. Besides that, we have to use a new ESP with the corrected code on the PCB. Further circuit testing and wire arrangement will help in ensuring it works.

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.

8.1 Future Work

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). We may replace the horizontal servo with a 360-rotating motor with built-in slip-ring to prevent wires from tangling. Additionally, we can replace the AC-DC converter with a 48 V battery. This allows us to make the robot modular, possibly adding wheels and remote control to move the robot around. Nevertheless, we believe that this work has good potential to be marketed as a polished product in the future.

References

[1] Meleap Inc., *Hado - techno sports*. Accessed: May 16, 2025. [Online]. Available: https://hado-official.com/en/.

[2] Kandidat, *Dodgeball object detection dataset and pre-trained model*. Accessed: May 16, 2025. [Online]. Available: https://universe.roboflow.com/kandidat-pgnqx/dodgeball-f7dp5.

[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, 2017.

[5] IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems, *Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems*, https://ethicsinaction.ieee.org/, IEEE Standards Association, 2019.

[6] ASTM International, *ASTM D1234-20: Standard Test Method for Example Testing*, West Conshohocken, PA: ASTM International, 2020. [Online]. Available: https://www.astm.org/D1234-20.html.

[7] Occupational Safety and Health Administration, *Guidelines for Preventing Workplace Violence for Healthcare and Social Service Workers*, https://www.osha.gov/sites/default/files/publications/osha3148.pdf, U.S. Department of Labor, OSHA 3148-04R, 2015.

[8] Association for Computing Machinery, *ACM Code of Ethics and Professional Conduct*, https://www.acm.org/code-of-ethics, Version 2.0, Approved June 22, 2018, 2018.

[9] Espressif Systems, *Esp-now wireless communication protocol*, https://www.espressif.com/en/solutions/low-power-solutions/esp-now.

[10] E. Systems, *Esp8266 arduino core - espnow.cpp source code*. Accessed: May 17, 2025. [Online]. Available: https://github.com/espressif/esp-now/blob/master.

[11] MessagePack Contributors, *Messagepack: It's like json. but fast and small.* 2025. Accessed: May 17, 2025. [Online]. Available: https://msgpack.org.

[12] I. Fette and A. Melnikov, *The websocket protocol*, RFC 6455, Dec. 2011. Accessed: May 17, 2025. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc6455.

[13] DS Servo, *Rds3225 servo motor datasheet*. Accessed: May 17, 2025. [Online]. Available: https://www.cytron.io/p-rds3225.

[14] Tower Pro, *Mg996r servo motor datasheet*. Accessed: May 17, 2025. [Online]. Available: https://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf.

[15] AZK Motor, *Ak300s bldc motor driver product page*. Accessed: May 17, 2025. [Online]. Available: http://www.azkmotor.com/ProductDetail/8679184.html.

[16] Microchip Technology, *Mcp41010 digital potentiometer datasheet*. [Online]. Available: https://www.microchip.com/en-us/product/mcp41010.

[17] Advanced Monolithic Systems, *AMS1117 Low Dropout Voltage Regulator Datasheet*. Accessed: May 17, 2025. [Online]. Available: https://www.diodes.com/assets/Datasheets/AMS1117.pdf.

[18] E. Systems, *ESP-WROOM-02 PCB Design and Module Placement Guide*. Accessed: May 18, 2025. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp-wroom-02_pcb_design_and_module_placement_guide_0.pdf.

[19] U. I. Forum, *Universal serial bus specification revision 2.0*, Section 7.2.1: Power Distribution Overview. [Online]. Available: https://www.usb.org/document-library/usb-20-specification.

[20] E. 4. C. Staff, *Ece 445 final report guidelines*. Accessed: May 18, 2025. [Online]. Available: https://courses.grainger.illinois.edu/ece445zjui/documents/ECE_445_Final_Report_Guidelines.pdf.

Appendix A Machine Vision R&V Table

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

Table 6: 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.	a) Set the computer to output "Candidate Anomaly" when a candidate anomaly is marked. 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. c) Record whether the system outputs "Candidate Anomaly". 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. e) Record whether the system outputs a "Candidate Anomaly". f) Perform at least 3 validations per group.
	2) Within a distance of 5 meters, a thermal anomaly pixel is classified as valid when the MCU returns the centroid position.	a) Prepare a tester to stand within five meters of the system and take at least five sets of images. 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. c) Record the number of times the system identifies the object. 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.

Table 6 – 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.	a) In the online environment, set the computer output the time at this point (T_1) after sending coordinate.
		b) Input an image with a valid object into the MCU.
		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).
		d) Make at least three measurements. The average time difference $\langle \lvert T_1 - T_2 \rvert \rangle$ should be less than 1 second.
	4) If no target is detected, the system should rotate the turret clockwise by 30° (±2°) and then initiate a new classification cycle within 1 s. REMOVED: We removed this test because we implemented a velocity tracker, which can detect fast moving targets.	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 V ± 3 V), Arduino analog input (A0) and GND (less than 1 V).
		b) Input a background image (i.e., no targets). The motor should rotate clockwise.
		c) Use the Arduino to read the analog value (0-1023) via `analogRead()`, map it to an angle (0°-300°), and record it.
		d) Take at least three measurements. The rotation angle should be between 28° and 32°.

Appendix B Firing System R&V Table

Table 7 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 7 shows the R&V table for both the turret and firing system module.

Table 7: 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 7 – continued from previous page

Subject Category	Requirements	Design Verifications
		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.
Projectile Reload Mechanism	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 Power R&V Table

Table 8 shows the *requirements and verification* table for the power module.

Table 8: 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) 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 1: 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_recv_cb(onReceive);
}

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

Listing 2: 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 9 shows the total expenses during the development of *dodgebot*.

Table 9: 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 (⌀3×200mm, 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

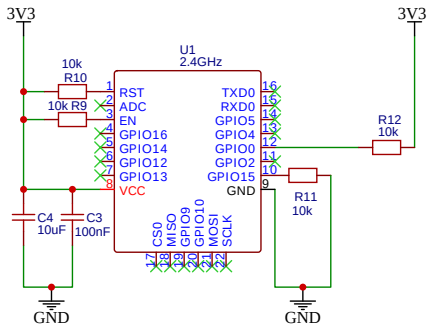
Table 10 shows the bill of materials (BOM) of the PCB design. Schematic and PCB design is shown in the next two pages.

Table 10: Bill of Materials (BOM) for the PCB design.

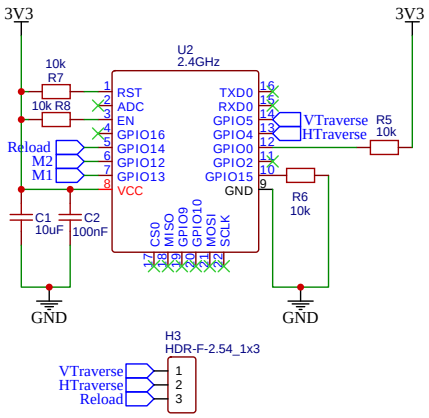
ID	Name	Designator	Quantity	Manufacturer	Supplier Part	Price (RMB)
1	10 μF	C1,C4,C5	3	Samsung	C15850	0.066
2	100 nF	C2,C3	2	Samsung	C1711	0.039
3	22 μF	C6	1	HRE	C6119868	0.25
4	F-2.54_1x3	H3	1	Ckmtw	C146690	0.56
5	F-2.54_1x4	H4,H5	2	ZHOURI	C5116530	0.48
6	M-2.54_1x1	6V,GND	2	BOOMELE	C81276	0.046
7	470 Ω	R1,R4	2	FOJAN	C2909361	0.003
8	150 Ω	R2,R3	2	YAGEO	C114523	0.013
9	10 kΩ	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

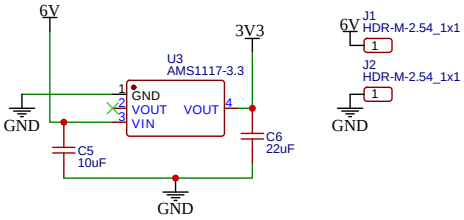
Master



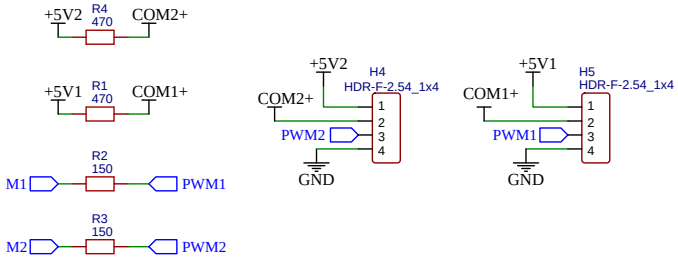
Slave



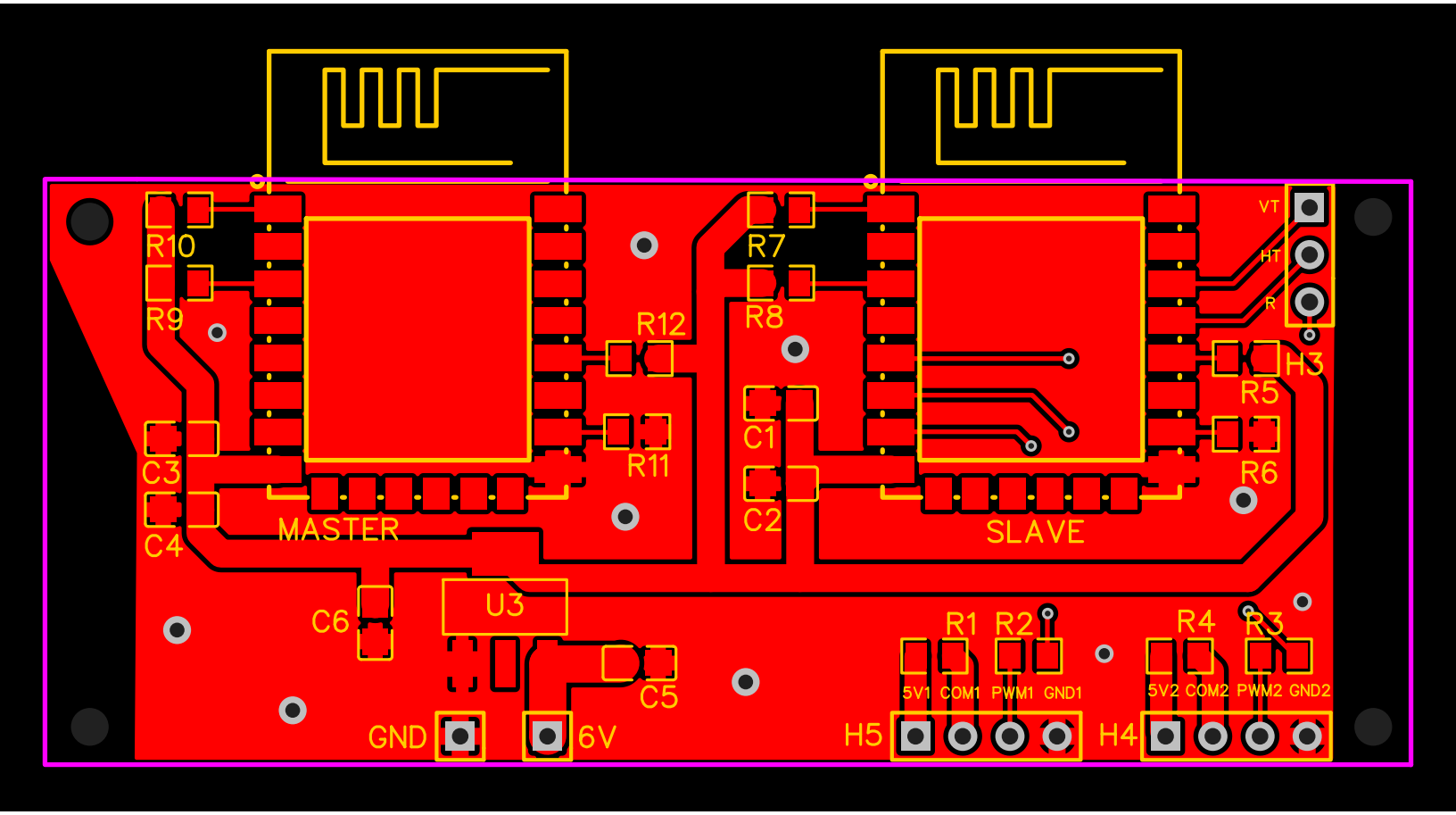
Power Conversion



48 V PWM Control



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


R10
R9
R12
R11
C3
C4
MASTER
R7
R8
C1
C2
SLAVE
VT
HT
R
H3
R5
R6
C6
U3
C5
GND
6V
R1
R2
R4
R3
5V1 COM1 PWM1 GND1
5V2 COM2 PWM2 GND2
H5
H4

Appendix G Base and Turret Schematic Design

Fig. 16 shows the initial turret design, which is an enclosure housing the firing system. Fig. 17 is the initial reload tube design. Fig. 18 is the initial magazine design.

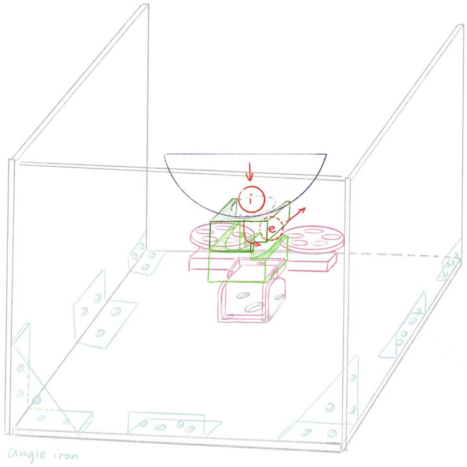


Figure 16: Initial design of the turret before changing it to the iron plate based turret.

Figure 17: Initial tube design.

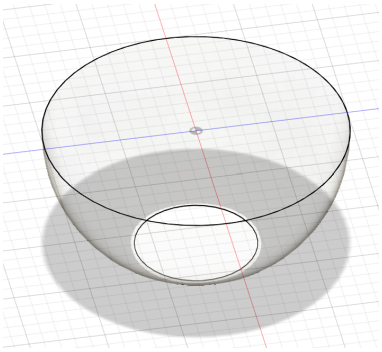


Figure 18: The initial bowl-shaped design.

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

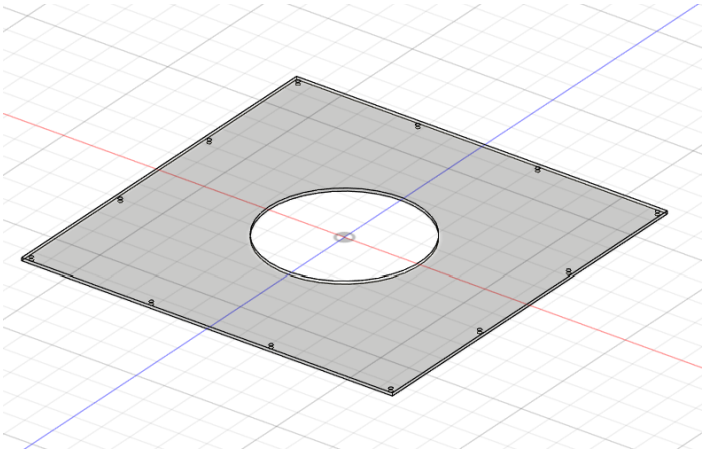


Figure 1. 3D model of the top cover for the base

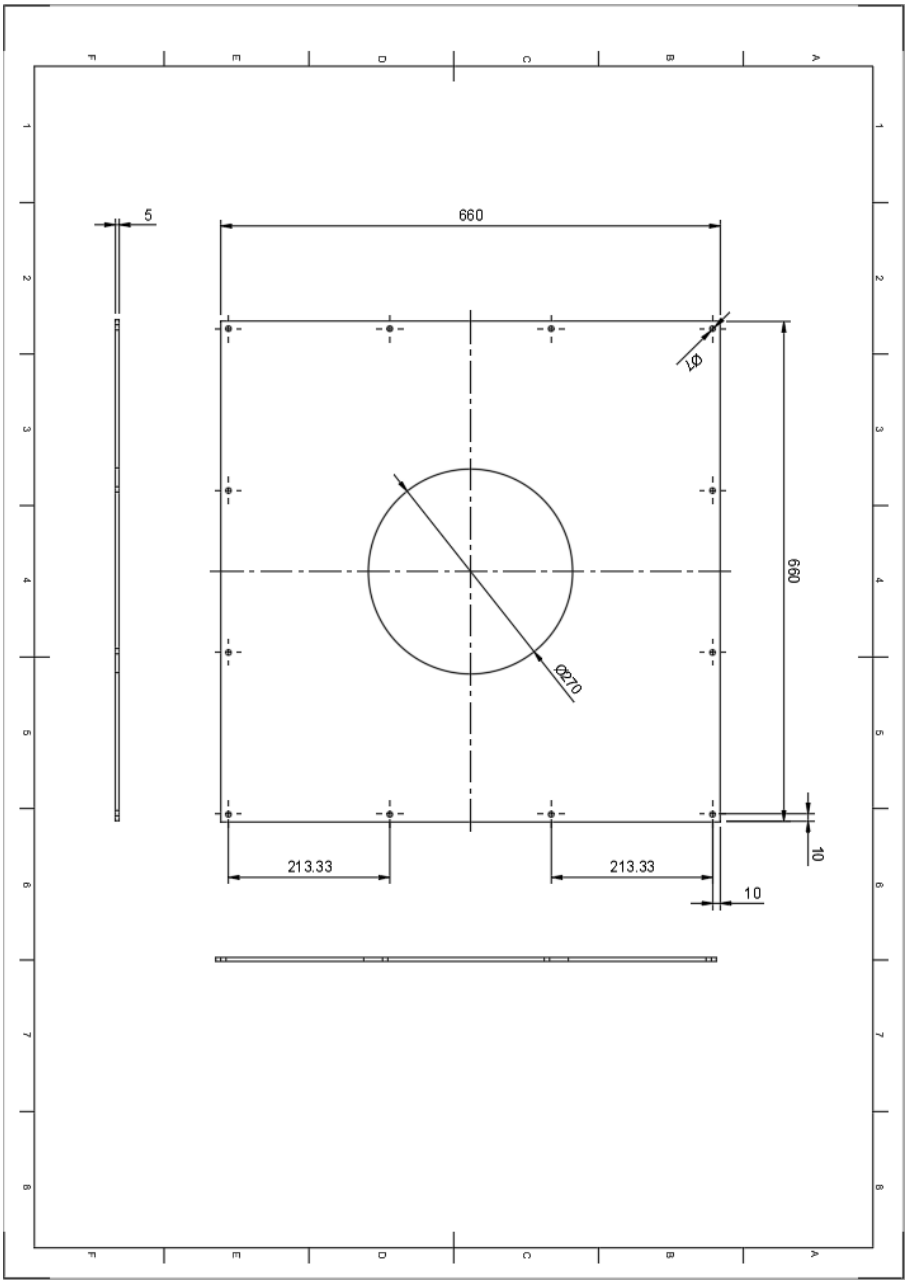


Figure 2. The engineering drawing of the top cover for the base

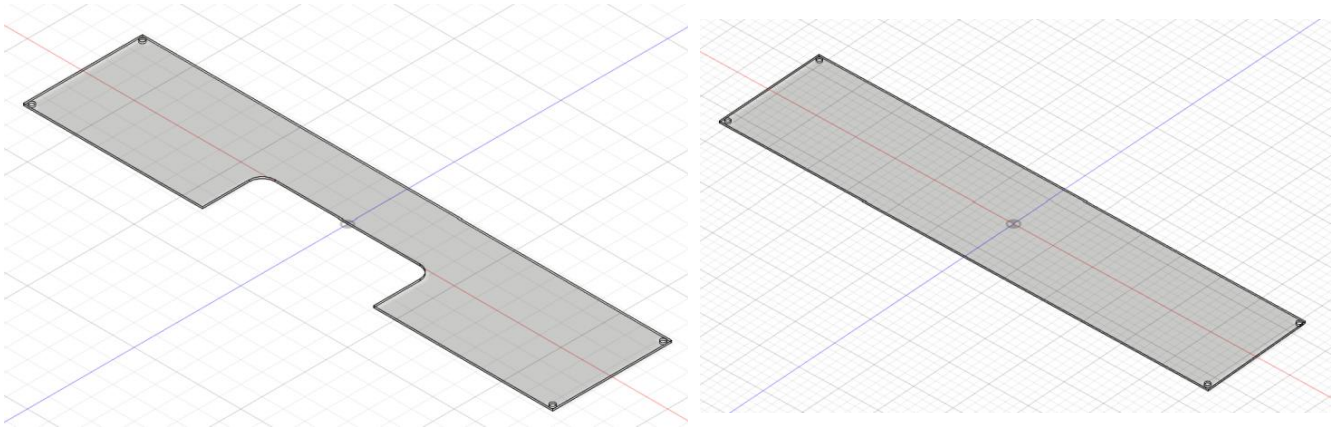


Figure 3. 3D model of the side wall with cable outlet (left) and without outlet (right) for the base.

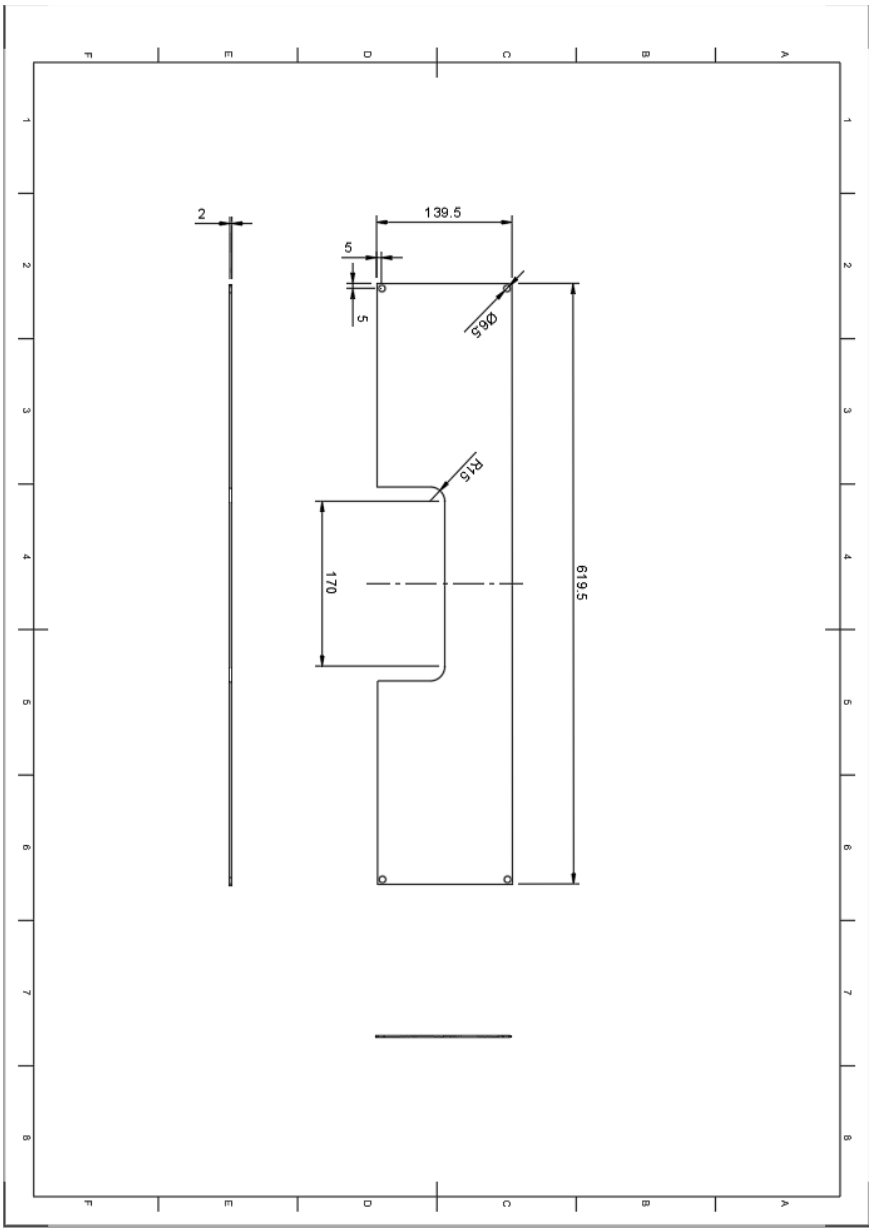


Figure 4. The engineering drawing of the side wall with cable outlet for the base

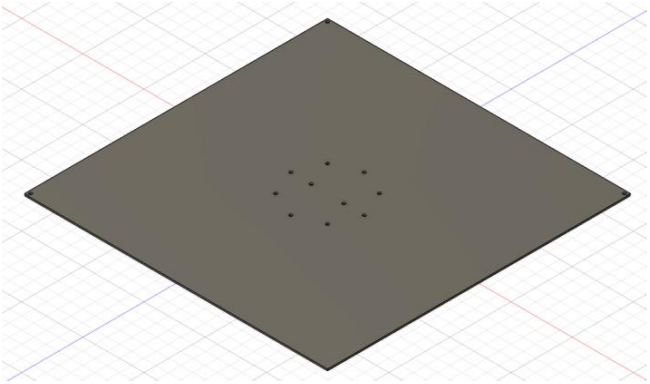


Figure 5. 3D model of the bottom board for the base

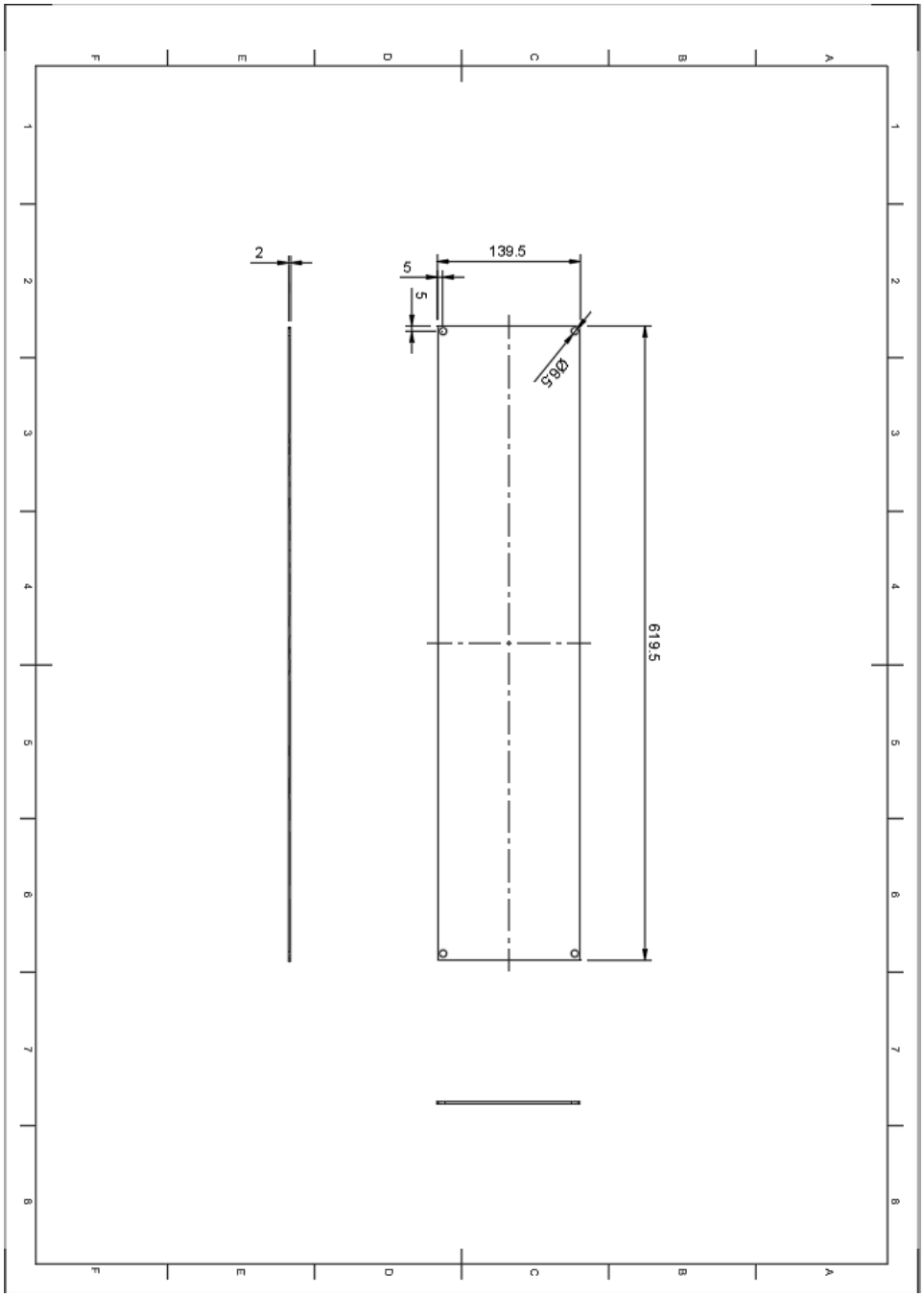


Figure 6. The engineering drawing of the side wall for the base

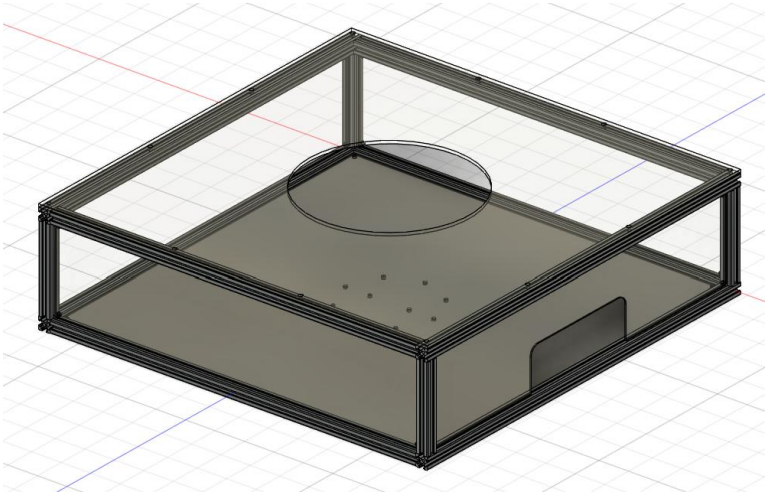


Figure 7. 3D model of the base with aluminum frame

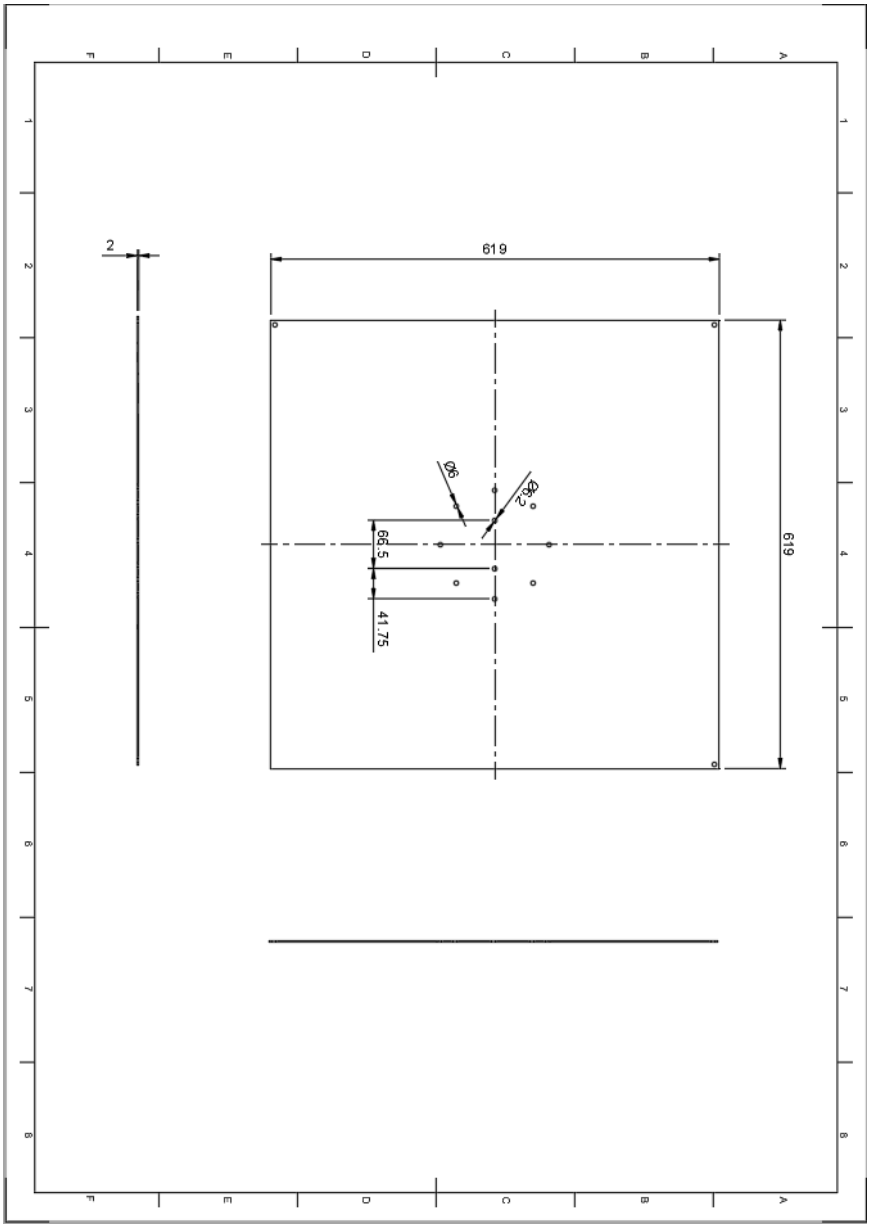


Figure 8. The engineering drawing of the bottom board for the base

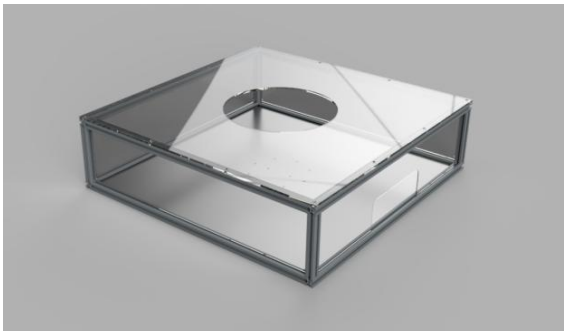


Figure 9. The rendering picture of the base with aluminum frame

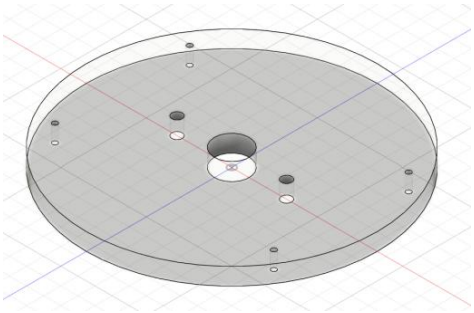


Figure 10. 3D model of the circular base for the gun

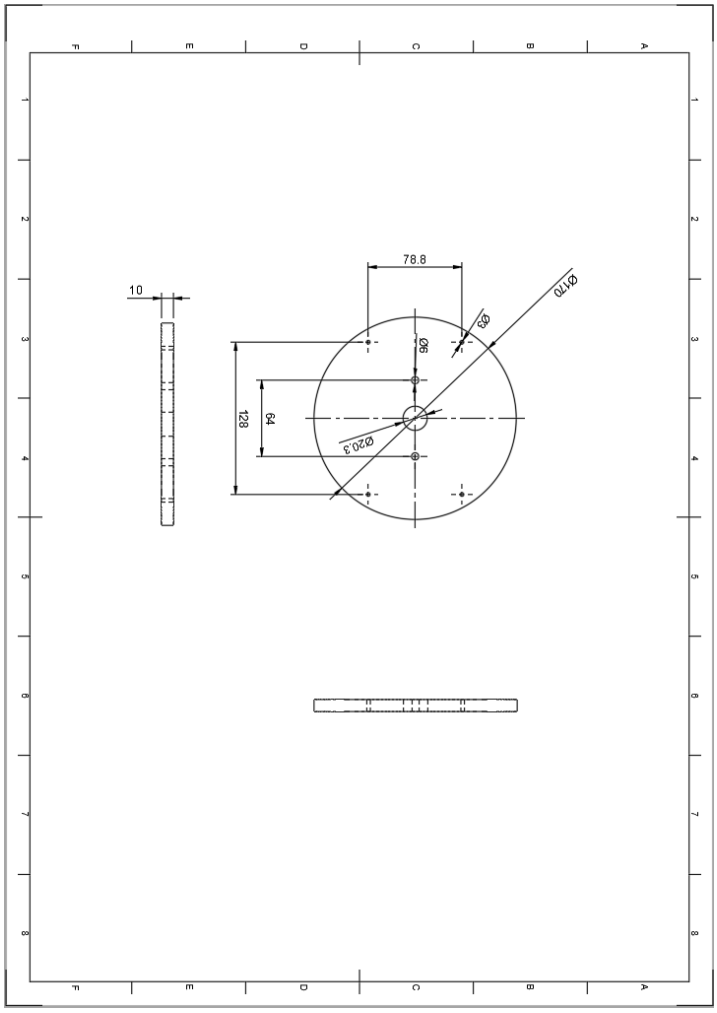


Figure 11. The engineering drawing of the circular base for the gun

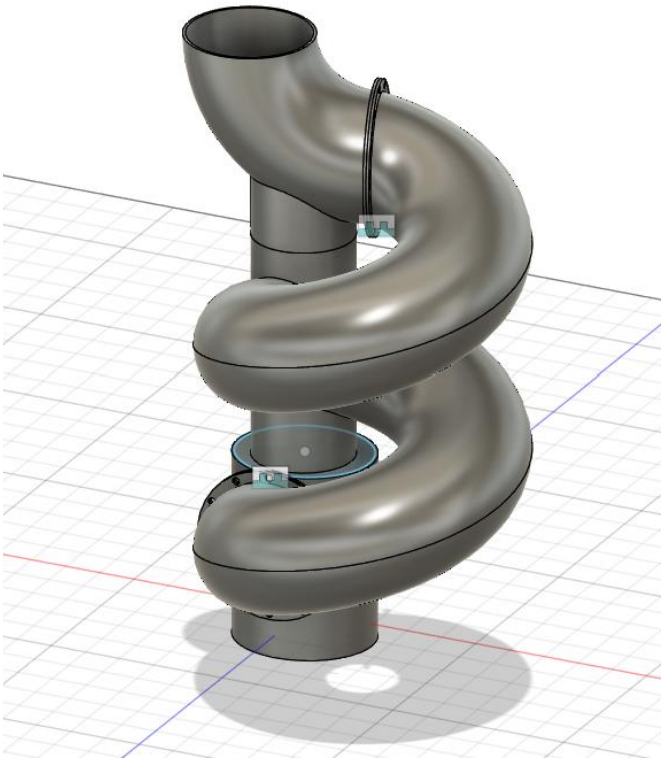


Figure 12. 3D model of the spiral pipe

Figure 13. The rendering picture of the spiral pipe

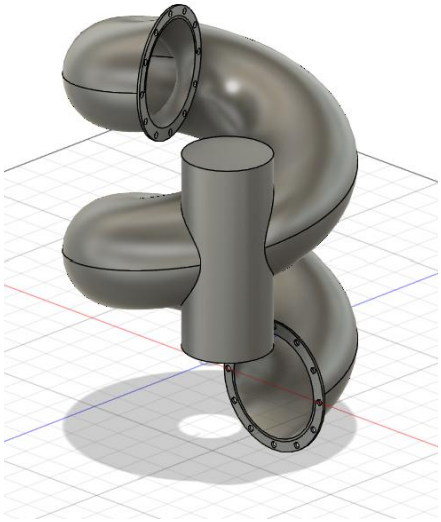


Figure 14. 3D model of the body of the spiral pipe

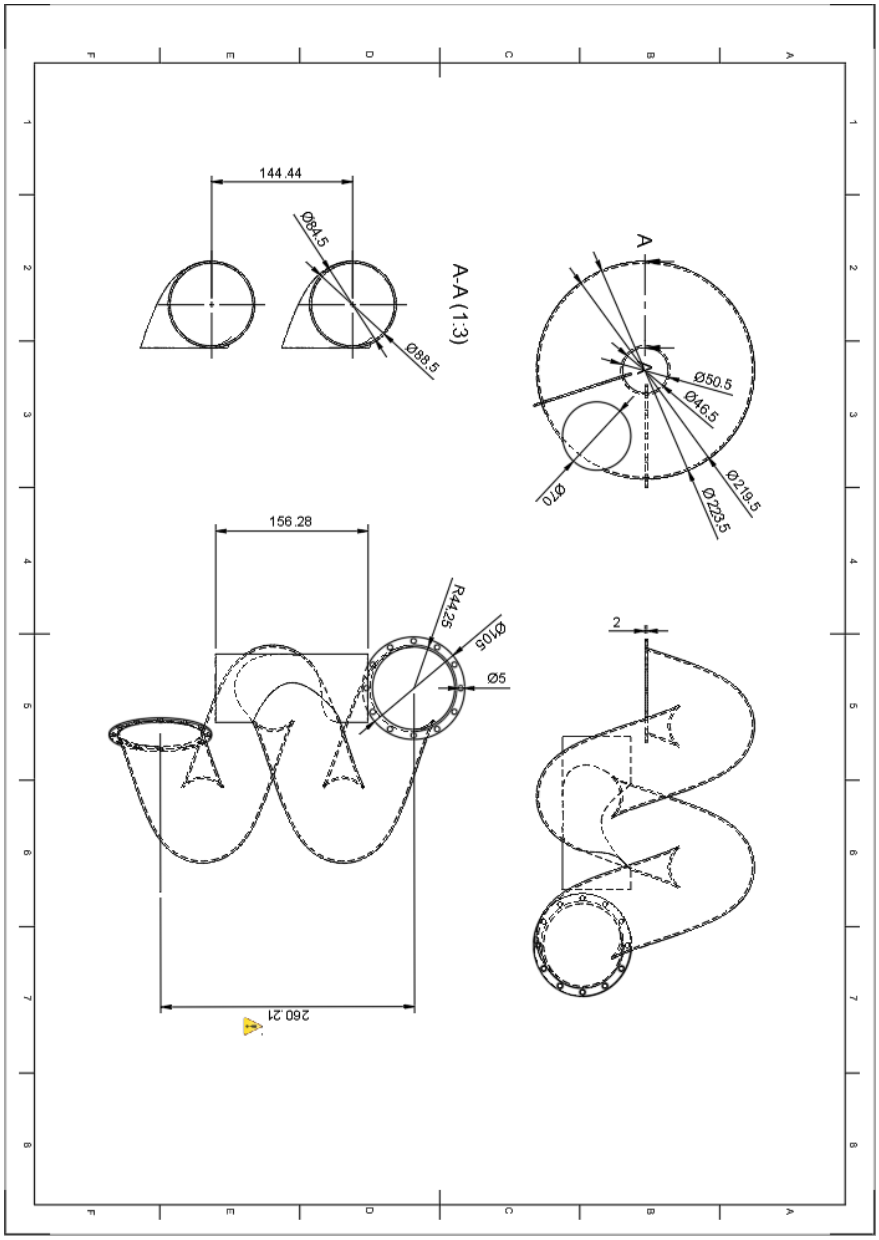


Figure 15. The engineering drawing of the body of the spiral pipe

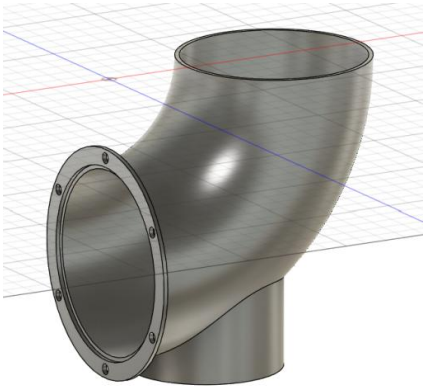


Figure 16. 3D model of the inlet of the spiral pipe

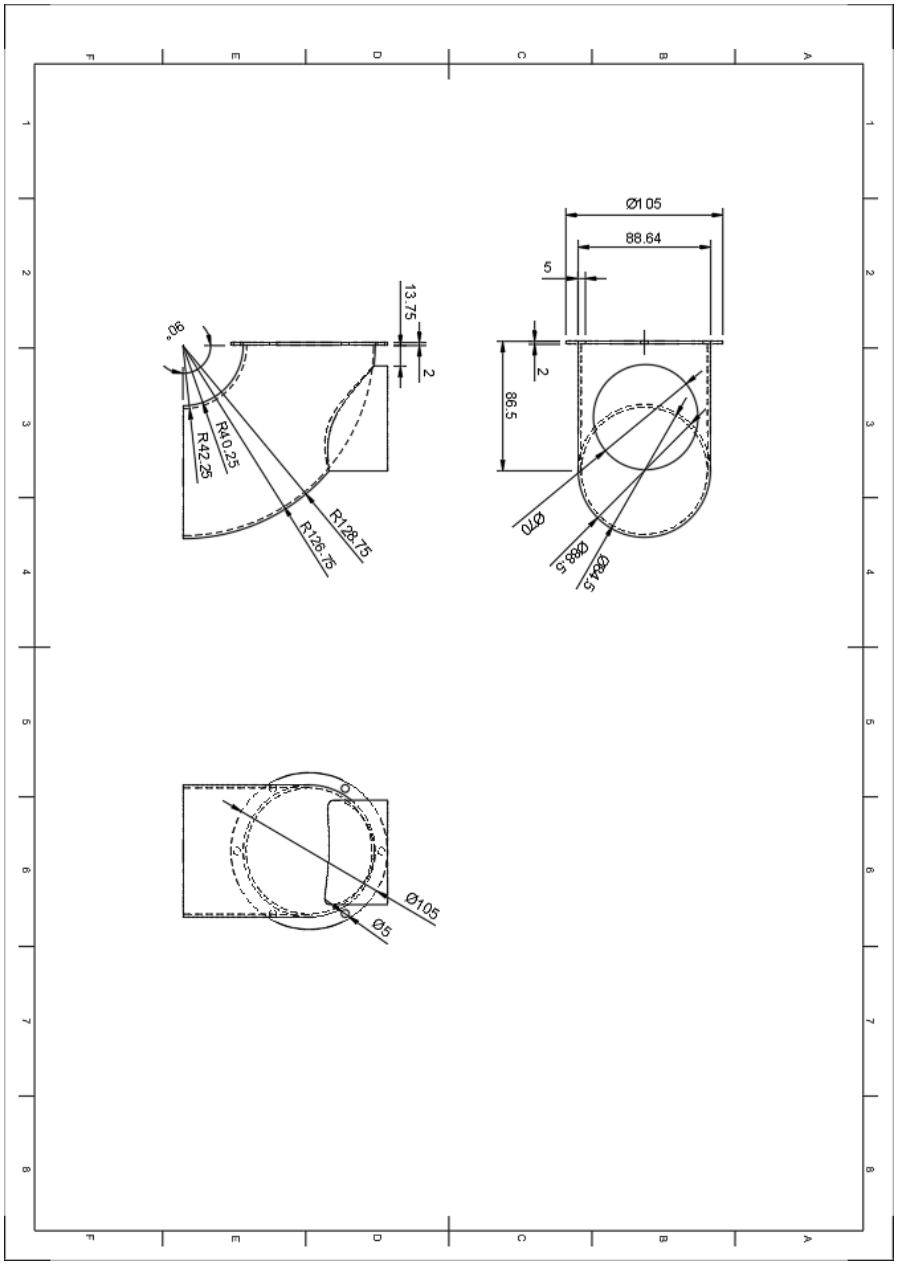


Figure 17. The engineering drawing of the inlet of the spiral pipe

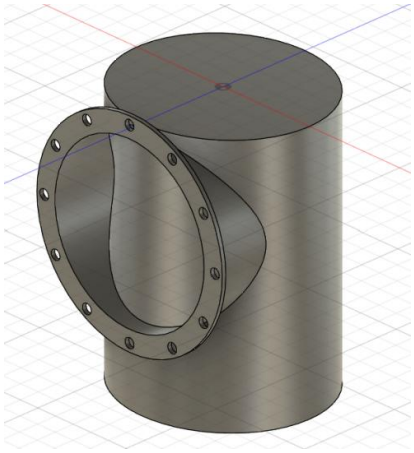


Figure 18. 3D model of the bottom end of the spiral pipe

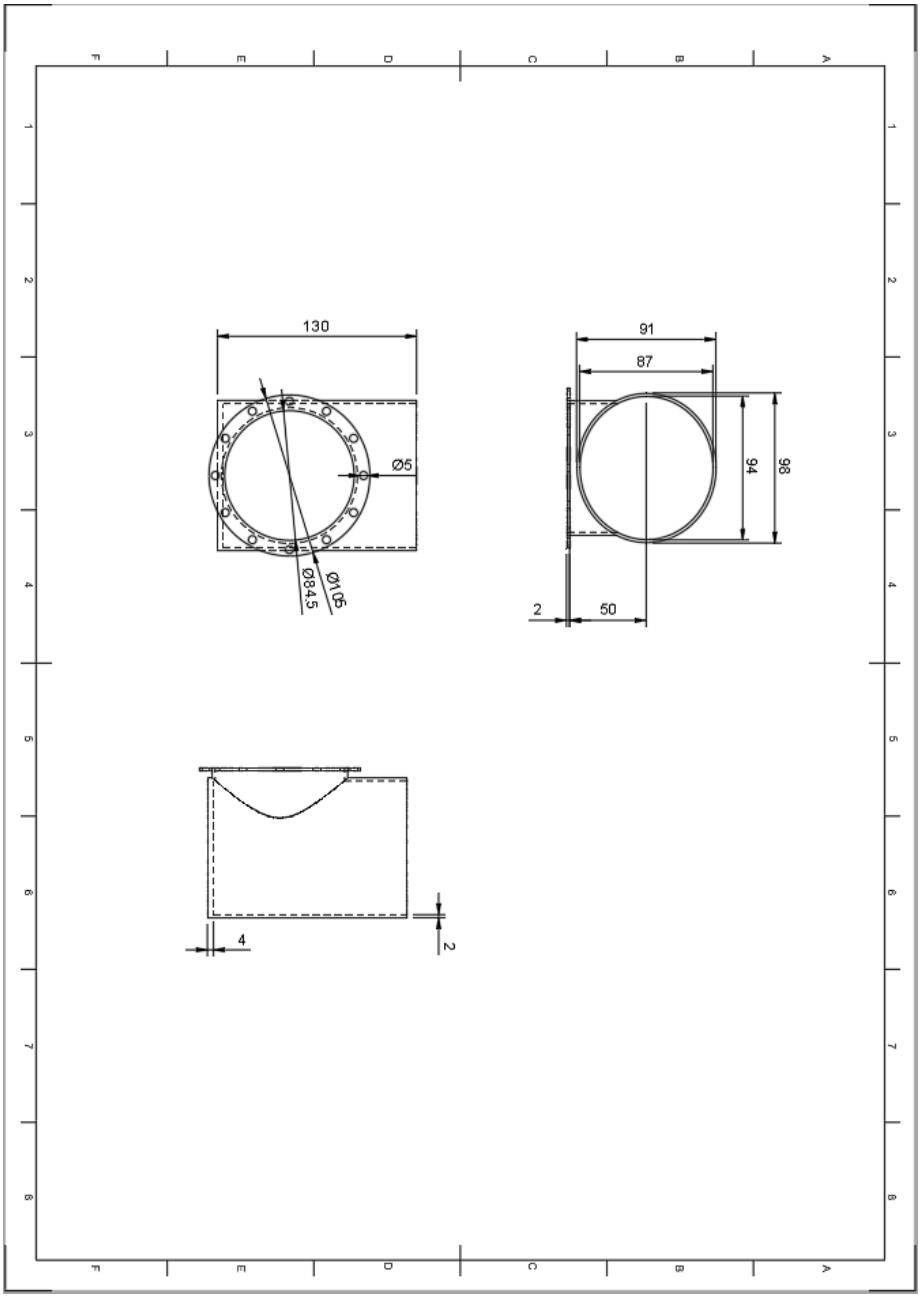


Figure 19. The engineering drawing of the bottom end of the spiral pipe