

Live Camera and Time-of-Flight Telemetry for an Ant-Weight Combat Robot

1 Matia Choi (hyunwoo)

2 Danilo Markus (dmarkus2)

3 Nikola Plavsic (plavsic2)

Final report Fall 2026 ECE 445, Senior Design

TA: Shengkun Cui

Abstract

We are building an ant weight combat robot for the ECE 486 Battlebots competition. The robot is built upon a custom PCB using an ESP32-S3 that streams 640x480 JPEG video from the camera and four flight of time sensors to the driver station over WiFi. The driver station takes the data and draws the proximity ring on top of the video feed;.

We target 20fps with no more than 150ms latency from the video feed and range accuracy of $\pm 50\text{mm}$. The robot starts the drive and weapon commands on board its microcontroller, so If the wireless link is intermittent mid match, the robot doesn't stop moving. We also implemented a hardware link-loss cutoff that kills motor power within 500ms if the link is fully disconnected. For

the best video feed result, we plan on creating a dedicated access point on the ESP32-S3 in the arena.

Contents

- 1. Introduction
 - 1.1 Problem
 - 1.2 Solution
 - 1.3 High-Level Requirements
- 2. Design
 - 2.1 Block Diagram
 - 2.2 Physical Design
 - 2.3 Subsystem Overview
 - 2.4 Subsystem Requirements
- 3. Tolerance
- 4. Ethics, Safety and Societal Impact
 - 4.1 Safety
- 5. Future Work
- References

Problem

This Project is an ant-weight battlebot under 2lbs for the Fall 2026 ECE 445 competition, where two robots fight in a plexiglass arena in the ECEB Atrium, controlled by a laptop driver station. We are scored on agility, control, and damage [1]. The rule also requires camera and flight of time sensor telemetry. Our robot needs to deliver these goals.

Solution

We plan on building a custom PC around the ESP32-S3 and 4 time flight sensors. The ESP32-S3 grabs the 640x480 JPEG frames from the on board camera and streams it over WiFi to the driver station. On the driver station, the time of flight sensors create a proximity ring overlay on the camera feed. The same WiFi link also sends the motor commands from the driver station to the robot. The robot will continue to drive when the video feed is slow or intermittent, however it will immediately shutdown if the link is completely lost.

The chassis and shell of the robot will be 3d printed with PETG. We chose PETG over PLA since it's strong but flexible enough to survive hits without cracking. We plan on printing the spinning disc in PA-CF filament. This filament is a high-performance 3D printing material made of nylon (polyamide) reinforced with chopped carbon fibers. This disc will be spun using a BLDC outrunner motor with a commercial ESC.

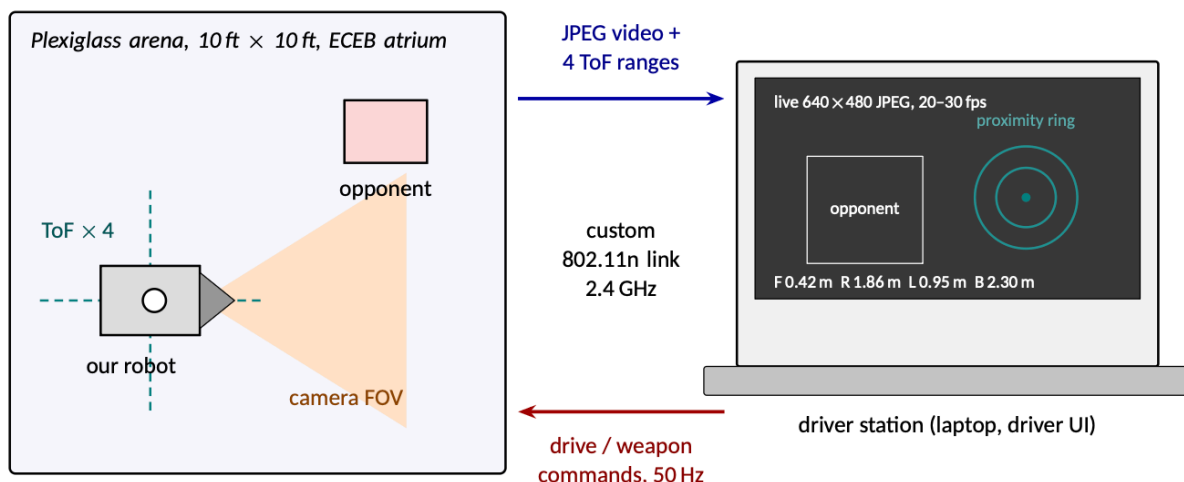


Figure 1

Visual aid of the robot in the arena and the data being streamed both ways.

High Level Requirements

1. The driver station needs to show a live 640x480 video feed from the robot with at least 20fps sustained for the full two minutes of the battlebots match in the ECEB atrium.
2. The driver station needs to draw the four time flight sensor data on top of the streamed video. Each within the range of ± 50 mm.
3. The robot needs to weigh under 2lb fully assembled with a battery. It needs to cut power from both drive motors and the weapon motor within 500ms of losing the connection to the driver station.
4. The driver station must allow the pilot to operate both drive motors and weapons from a laptop with commands sent over WIFI with a maximum latency of 100ms.

2. Design

2.1 Block diagram

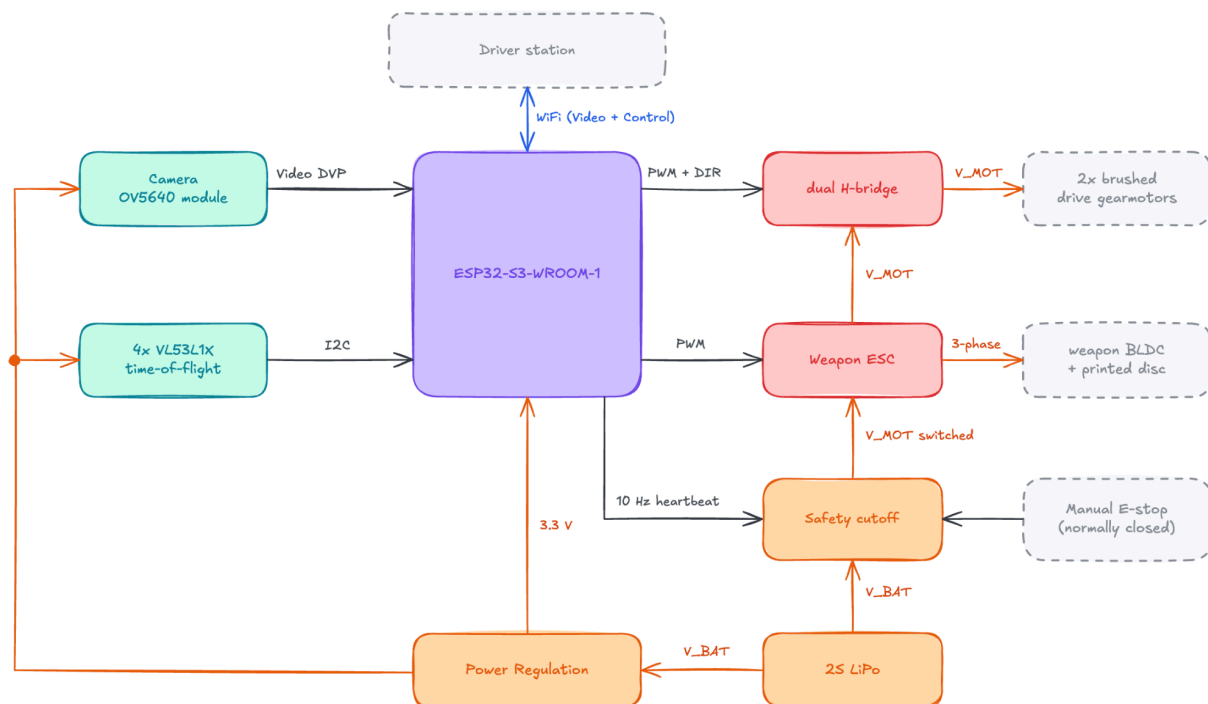


Figure 2: made with Excalidraw

Figure 2 shows the whole system. It shows the Camera and the 4 flight of time sensors, drive motors, weapon esc, power system, and the driver station.

There are 5 main connections to the ESP32-S3, DVP data line for the camera, i2c line for the 4 time of flight sensors, 2 pwm signals for the drive motors, a pwm signal for the weapon motor, and a wireless link over WiFi to the driver station.

There are two main power rails. Logic power rail for the microcontroller and the sensors, and a power rail for the motor systems. During an emergency/manual shutdown, the motor power rail will shutdown.

2.2 Physical Design

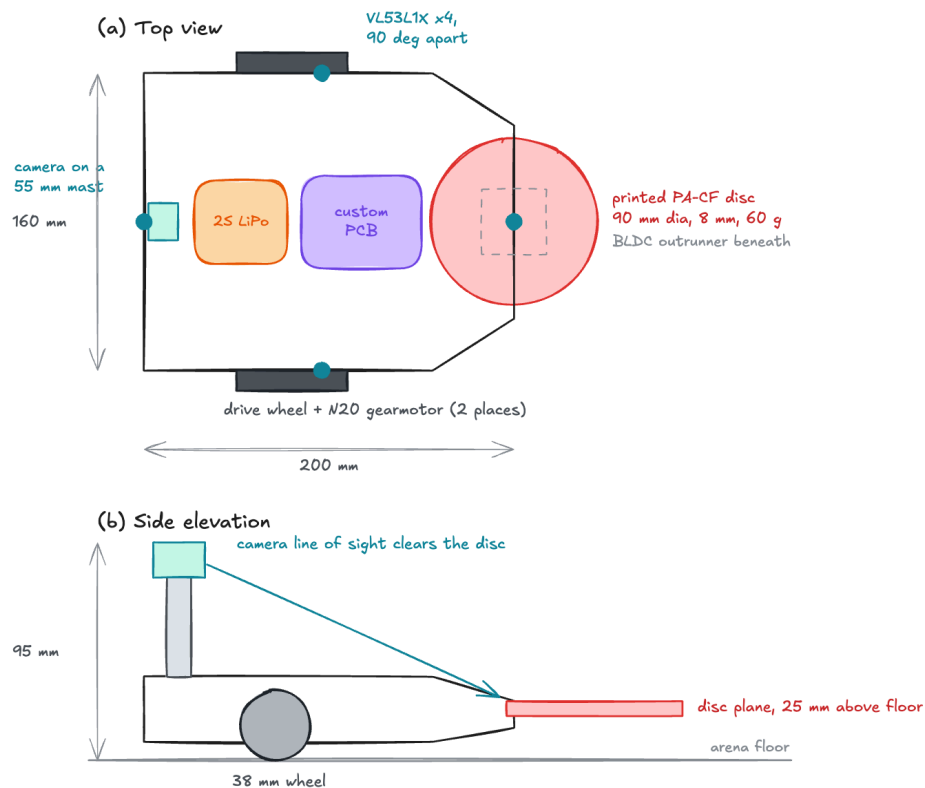


Figure 3: made with Excalidraw

Figure 3 shows the mechanical design of the 3d printed robot. The camera sits on the rear, angling down such that it can see over the rotating disc. The 4 flight of time sensors are placed 90 degrees of one another over each side of the robot.

Item	Mass (g) Estimate
Printed chassis, shell, mast (PETG)	250
Drive gearmotors, 2 × N20	70
Wheels, 2 × ø38 mm	20
Weapon BLDC outrunner	32
Weapon disc, printed PA-CF	60
Weapon ESC, 30 A	10
Custom PCB, assembled	25
Camera module and mast hardware	12
4 × VL53L1X and harness	12
2S LiPo, 850 mAh	50
Fasteners, wiring, E-stop switch	40
Total	581
Margin to limit	327

Table 1

Shows the mass budget against the 2lb (908g) inspection limit

2.3 Subsystems

2.3.1 Subsystems and Requirements

Controller System

The robot uses an ESP32-S3 as the main controller. It acts as the main processor and a communication device for the robot. The robot uses 2.4 GHz Wi-Fi connection for communication with the driver station. It uses the DVP interface to communicate with the OV 5640 camera. It has four VL53L1X time-of-flight sensors that are driven by the I2C bus. The ESP32S3 also generates the PWM signals required for both drive motors on each side of the robot and the weapon motor ESC.

During the operation of the robot, the controller continuously collects data from the four VL53L1X time-of-flight sensors and the OV5640 camera. It sends this data to the driver station using the aforementioned Wi-Fi link. At the same time, it also receives drive motors and weapon motor control outputs. A short temporary loss in the video link does not stop the robot. However,

if the connection is completely lost, it automatically shuts down all power systems for the robot and disables itself.

C1	Radio Link	Power system shuts down when radio link is loss within 500ms	Run the robot for 2mins. During the run, disconnect the wifi, or move into a different room to purposefully disconnect the link. Verify that the robot auto shutdown.
----	------------	--	---

Sensor System

The sensor subsystem uses an onboard OV5640 camera module to provide a live view from the robot. The ranging subsystem uses four VL53L1X time-of-flight sensors placed around the chassis to measure nearby obstacles and provide distance information to the driver. The sensor system is designed such that it streams 640x480 JPEG video streams at an average fps of 20.

The four VL53L1X time-of-flight sensors are placed approximately 90 degrees from one another around the chassis such that the driver can get information about their surroundings without relying on the video feed. The ToF sensors communicate with the ESP32-S3 using the I2C bus. The ESP32-S3 will periodically read from each of the devices through the I2C bus ensuring that the measured distance is within 50mm of the actual distance.

S1	Camera / Video	Provides 640×480 JPEG video at an average frame rate of at least 20 fps for 2 minutes	Run the robot for 2minutes. Record average FPS every 10 seconds. All measurements must be above 20fps.
S2	ToF Ranging	Each of the 4 ToF sensors must return accuracy within 50mm.	Run the robot, and place out a block on one side of the robot and set the distance from the robot to 50, 100, 150 and 200mm. Then we repeat for all four sides. They must all be within 50mm of the target.

Motor System

The motor sub system consists of 2 N20 brushed DC geared motors for driving the robot and a brushless BLDC motor for spinning the weapon disc. The two N20 brushed DC geared motors

are placed on each side of the chassis and are responsible for maneuvering the robot by controlling the speed of each wheel. The drive motors are powered by a dual channel MOSFET H-bridge TB6612FNG. The PWM signal is sent from the ESP32-S3 to control the drive motors. The weapon motor driver, a 20A hobbywing BLDC driver, also receives a PWM signal from the ESP32S3.

Each motor driver receives power from the power system. A separate hardware safety cutoff system is in place for the motors. There is a MOSFET in series with the power to the motor drivers. If there ever is a radio link loss for more than 500ms, the ESP32 shuts down the MOSFET, cutting all power to the motors.

M1	The Motor responses to the driver command within 100ms.	Monitor the motor control pwm output using an oscilloscope while sending a command from the driver station. Measure the delay between pressing a key on the driver station to seeing a corresponding PWM signal on the oscilloscope. Repeat it at least 10 times. All measurements must be under 100ms.
----	---	---

Power system

The power system uses a 2S Lithium Polymer battery as the main power source for the battle bot. The battery supplies both low voltage used by the sensors and the ESP32-S3 and high voltage used by the two drive motors and the weapon ESC. The low voltage is regulated through a voltage regular into a separate low voltage logic rail while the motor power rail is a separate high voltage line that powers the drive motor esc and the weapon motor ESC.

The logic power system must maintain a stable supply to keep the ESP32-S3, the time of flight sensor, camera, Wi-Fi link, the drive motors, and the weapon motor running simultaneously. Separating the power rail with the logic rail allows us to cut the power off to the motors without losing radio and telemetry.

P1	The power system must maintain a stable supply to keep all systems operational	Fully charge the battery. Start the 2min robot run. Turn the weapon motor on, and drive around. Confirm that all systems are operational.
----	--	---

Driver Station

The driver station is software within the laptop that holds the robot control interface. From here the pilot should see a live feed, proximity and transmission information and communicates piloting and weapon operation commands to the robot over WIFI.

D1	The driver station sends piloting and weapon commands from the laptop to the robot at a maximum latency of 100ms.	Using an oscilloscope or slow-motion footage with a timer measure the time from pc input to robot action output. Measure both weapons and driving response, and do several (10) trials. Failed if any measurements fall below 100ms.
----	---	--

2.3.2 Parts and Labour cost

A rough cost estimate for the major components is:

ESP32-S3-WROOM-1: about \$8

OV5640 camera module: about \$10

Four VL53L1X time-of-flight sensors: about \$20

TB6612FNG motor driver: about \$5

20 A weapon ESC: FREE matia has one

Two N20 drive motors: about \$15

BLDC weapon motor: about \$15

2S LiPo battery: about \$15

Voltage regulators and power components: about \$10

Safety cutoff components, MOSFET, and E-stop: about \$10

Custom PCB fabrication and assembly: about \$20–30

Wiring, connectors, headers, and miscellaneous electronics: about \$10

Wheels and small mechanical hardware: about \$10

3D-printed chassis and armor material: about \$10–20

The complete robot is therefore expected to cost roughly \$175–200, excluding the laptop and reusable lab equipment.

Labor

Assume each team member is paid \$22/hour as high-skilled interns and works approximately 120 hours on the project.

$\$22/\text{hour} \times 2.5 \times 120 \text{ hours} = \$6,600$ per 3 group members = \$19,800

This makes parts + labor work out to \$20,000 assuming full expenses paid for parts.

Schedule

The following is a drafted schedule where the listed requirement is the group member's focus area. On all substantial parts of the project extensive collaboration is expected so that work is spread evenly and no one is in the dark about any system or aspect of the finished bot.

Week (Starting from today)	Matia	Danilo	Nikola
1	Finalize sensor/camera architecture	Finalize mechanical architecture	Finalize controller/power architecture
2	Order sensor/camera parts	Design chassis/weapon	Design PCB schematic
3	Develop sensor interface	Print chassis prototype	Complete PCB layout
4	Develop video streaming	Assemble drivetrain/weapon	Bring up ESP32/power system

5	Integrate ToF sensors	Refine mechanical assembly	Implement motor/Wi-Fi control
6	Integrate telemetry	Assemble final chassis	Integrate PCB/safety cutoff
7	Test camera/ranging	Test drivetrain/weapon	Test control/link-loss cutoff
8	System testing	System testing	System testing
9	Final verification	Final verification	Final verification
10	Final report/demo prep	Final report/demo prep	Final report/demo prep

3. Tolerance

Three design consequences drop out of $C_{min} = 4.92$ Mbps.

The transport is supposed to drop frames, not queue them. Over TCP, a channel slower than the offered load keeps growing the send queue indefinitely. Latency grows, offering a smooth stream of video that is increasingly outdated, which is not useful compared to the most recent information about our location and the actions of our opponents. We therefore choose to send over udp with at most one frame in flight and and throw away any information less current than the most recent frame. The channel will always choose to compensate by losing frame rate in place of adding delay. Table 3 is the design's supposed degradation curve following these variables.

To accomplish this we will require a quality controller. We will prioritize trading S for frame rate and adjust a quantizer to keep the in-flight time around one frame period. At QVGA fallback (res=320x240, S 10kb, 20fps) the load will fall to 1.6 Mbps to give us best-case control in the worst channel that we're likely to encounter. A 20 MHz 2.4 GHz channel gives 15–20 Mbps of application goodput, so $C_{min} = 4.92$ Mbps could brace about 70% of airtime being taken by other traffic before the frame rate requirement breaks.

A ranging read is circa 20 bytes including register addressing, which is roughly 190 clock cycles including framing; At 400 kHz gives 0.48 ms. 4 sensors x 50 Hz yields 200 transactions per second, giving 9.5% bus utilization. This means that the shared bus doesn't interfere with the 40 Hz-per-sensor requirement. The worst-case age of a range sample when a frame renders is one ranging period plus transport; inside the 100 ms alignment requirement of high-level requirement 2.

For a disc weapon of mass $m = 60$ g and radius $r = 45$ mm modelled as a uniform disc,

$$I = \frac{1}{2}mr^2 = 6.1 \times 10^{-5} \text{ kg} \cdot \text{m}^2, \quad E = \frac{1}{2}I\omega^2 = 48 \text{ J at } 12,000 \text{ rpm.} \quad (4)$$

At 7.4 V with a 15 A current and around 60% of electrical power reaching the disc as kinetic energy, $t = E/\eta P = 48/67 = 0.72$ s, less than half of the 1.5 s allowed by requirement A2. The 48 J figure is what drives the safety component of Section 3.2.

The monostable period is given by $t = 0.45RC$ with $R = 2.2 \text{ M}\Omega$ and $C = 0.33 \text{ }\mu\text{F}$, $t = 327$ ms. The worst possible delay is 427 ms(100 ms link-loss detection window plus the monostable period). Allowing give or take 10% on both R and C gives a worst case of about 460 ms, inside the 500 ms of high-level requirement 3. This is admittedly close and we can verify timing on the physical prototype to adjust.

5. Ethics, Safety and Societal Impact

The first item of the IEEE Code of Ethics is the commitment to “hold paramount the safety, health, and welfare of the public” [2]. In order to take care of public safety, the weapon will be off at power-on, the motor cutoff will be hardware based so that a bug cannot threaten safety, and we won't activate the weapon anywhere that risks the safety of others.

The same section includes a commitment to be honest and realistic in stating claims [2]. All this

means is that we should be honest that every number in Section 2.4 is an estimate. We will report our numbers in the final document as we measure them and record honestly when we fall short of our own expectations.

The robot also carries a camera into a public atrium. The video is streamed point-to-point to the driver's laptop over our own access point, it's not recorded or retained beyond the match. In this scenario we believe that we meet a standard of reasonable privacy given that attendees are aware that the robots will use cameras in some capacity.

Safety

From equation (4), the disc could use about 48 J at 12,000 rpm making it the most dangerous aspect of the bot. The way we address this is as follows: the disc only spins inside the arena or a closed containment fixture since the weapon is disarmed by default at power-on and needs an explicit arm command, and nobody will touch the robot until the connector is physically removed (not just when software says disarmed). The same will apply in testing.

A robot that keeps driving or attacking is the primary danger of the competition we are required to prevent [1][3]. See hardware cutoff from Section 2.3.4, with timing verified in Section 2.5.3 at a worst case of about 460 ms against the 500 ms requirement. Every failure we can imagine from the provided design would end in the motor turning off.

The battery pack could also be dangerous outside the arena, so we'll make sure we store it at storage voltage and keep it away from fire hazards and not charge unattended. We will make sure to dispose of damaged or dysfunctional batteries in the manner mentioned in our required safety training.

We will follow the lab safety guidelines in the Senior Design Lab safety assignment and print on the lab walls and manuals in all tasks we do.

Future Work

Future work could use this platform beyond combat robotics to inform teleoperation for dangerous inspection and disaster-response applications. Because the design relies on cheap parts, a single microcontroller, and 3D-printed body, it could also serve as an affordable robotics plan in education. In general, our project provides a practical way to teach real-world constricted power electronics, RF communication and embedded systems that many students only see in streamlined homework problems until they enter the workforce.

Climate considerations could also be a direction to look towards. Replaceable armor panels already limit the amount of material discarded after damage, however we could think more about battery choices and eco-friendly materials.

References

- [1] V. Gruev, "Ant-weight, 3D printed Battlebots competition," ECE 445 course materials, Univ. of Illinois Urbana-Champaign, Fall 2026.

- [2] IEEE, "IEEE Code of Ethics," IEEE Policies, Section 7.8, Jun. 2020.
[Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>
[Accessed: Sep. 18, 2026].

- [3] National Robotics Challenge, "Robot combat — ant weight class rules."
[Online]. Available: <https://www.thenrc.org/> [Accessed: Sep. 18, 2026].