Prototype Design and User Manual
Indoor Line Following Drone

Table of Contents

List of Figures

List of Tables

1. Introduction

The manual is designed to provide comprehensive guidance on the design, assembly , setup and operation of the indoor line following drone, developed from Darwin Baby Ape 3” drone, which is exclusively built for autonomous remote safety inspections of the Australian National Fabrication Facility(ANFF) laboratory.

The drone model is developed from Darwin Baby Ape pro-3” toothpick drone kit which is equipped with advanced features like AIO-FC. The prototype is built to meet the specific requirements as stated by the stakeholders.

The methods and techniques outlined in the drone is specifically tailored to the upgraded version of Darwin Baby Ape Pro 3” drone, configured for line-following specifications. This manual only provides the basic setup and operational guidelines required to build and test the drone.

While the system encountered limitations in achieving a successful flight during testing, the approaches described here can serve as a foundation for further research and improvisations. The information presented is intended to assist in the refinement of the current prototype and to inspire future enhancements in drone design and functionality.

2. Product Specifications

Table 1: Drone specifications

Drone Kit used	Darwin BabyApe Pro V2 FPV Drone
Frame structure	3-inch toothpick frame
Flight Controller	F411-AIO FC
Motor	1104 4300KV 2-3S Brushless Motor
Recommend Battery	1S-2S LiPo battery
Propeller	Gem fan 3016 propeller
Battery	GNB 720mAh, 2S 100C battery
Flight time	7 mins
Camera and Microcontroller	ESP32-Cam
Object Detection	Up to 30 cm
Charger	MMOBIEL USB Charger Cable
Weight	130 g

Table 2: Electronics Components List

No	Description	Specification	Qty
01	Flight Controller	F411-AIO FC	1
02	ESP32-Cam	ESP32 Wi-Fi-Bluetooth Camera Module Development Board	1
03	Camera	OV2640 Camera Module	1
04	Ultrasonic Sensor	HC-SR04	1
05	External RF Antenna	RF Dual Band RP-SMA Antenna	1
06	Battery	GNB 720mAh, 2S 100C battery	1
07	Receiver	ELRS Re1ceiver	1

Table 3: Mechanical Component List

No	Description	Specification	Qty
01	BLDC Motor	1104 4300KV Motor	4
02	Propeller	Gem fan 3016 Propeller	1
03	Screw	M1 4*5	20
04	Screw	M2*4	5
05	Screw	M2*6	10
06	Screw	M2*8	5

07	Screw	M2*10	5
08	Screw	M2*12	5
09	Nuts	M2 nylon nuts	15
10	Nuts	M2 Iron nuts	5
11	Pads	M2*1mm*0.5mm	5
12	Columns	M2*15	5

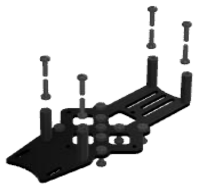
Table 4: Accessories List

No	Description	Specification	Qty
01	ESP32-CAM mount plate	3D printed model for mount plate	1
02	Propeller Guard	3D printed propeller shrouds	4
03	Landing Gear	3D printed landing gear	1
04	Ultrasonic Sensor Holder	3D printed model -Ultrasonic Sensor holder	1
05	Jumper Wire	Connecting wires	1 pack
06	Adhesives	Hot glue gun, double sided tape	1 each

3. Part 1- Hardware Integration

3.1 Frame Built

The Drone kit comes with a 3” frame consisting of top, mid plate and bottom plate along with crush plate and a battery strap. All the frame plates are to be assembled using the mechanical components starting from No 3 as listed in table 3. The top, bottom, mid and the assembled figure is shown below:



(a)

(b)

(c)

(d)

Figure 1: Assembled (a) bottom plate (b) top plate (c) arms (d) battery strap

3.2 Drone Control Unit

The drone control Unit consists of an F411-AIO Flight Controller along with an STM32F411CEU6 microcontroller as the main processing unit.

Figure 2: F411-AIO FC

The key features and instructions for usage are detailed below:

1. Overview and Key Features:

- Processor: The FC is integrated with an STM32 F411 microcontroller that enables efficient data processing in real time.
- A built-in 6-axis gyroscope and accelerometer that helps in real-time stabilization thus enabling stable flight.
- Input Voltage: 3.7 V
- Constant current: 15 A

2. Power & Connectivity

- Power Distribution: Supports 1S to 4S LiPo batteries
- ESC Compatibility: Supports both BLHeli_S and DShot protocols thus enabling smooth and efficient motor control.
- Receiver Compatibility: Supports SBUS, IBUS, DSMX, and other popular radio receiver protocols.

3. Setup & Configuration

- FC F411-AIO is configured using Betaflight Configurator. This allows firmware updates, fine-tuning flight parameters, and monitoring sensor performance.
- The Betaflight firmware is flash-updated to the flight controller using a USB cable.
- The FC is flashed by altering the setting in the Betaflight configurator
- The blinking of the LEDs indicates that the FC is flashed.
- The FC can also be flashed by connecting the battery thrice and then finally connecting the battery completely to the system.
- The FC also enables PID tuning to optimize the drone’s response.

4. Installation

- The F411-AIO is compactable and perfectly fits inside the drone.
- It is mounted to the frame using screws

- The flight controller is then wired to the motors, ESC, and other components depending upon the wiring diagram.

5. Maintenance

The FC must be handled with care to avoid damage and to ensure maximum performance.

- The FC should be properly ventilated to avoid overheating during operation.
- The FC must be updated periodically to have access to the latest features and to have maximum performance efficiency.
- The FC must be properly secured to the frame to protect it from any physical damage in case of any crash landings.

3.3 Motor and Propeller

The propulsion system includes a set of four motors along with four propellers.

3.3.1 Motor

Figure 3: 1104 4300KV Motor

The motor used in the drone kit has the following specifications:

1. Overview of the motor:

- Model: 1104 4300kV Brushless Motor
- Application: Mainly designed for small, compact light weighted drones like toothpick drones and micro-quads
- Suitable Battery: Mostly used along with 1S or 2S LiPo batteries. Can be used with 3S batteries only if proper cooling techniques are provided. Ideal for 2S batteries.
- KV Rating: The rating of the motor used is 4300kV which means that the motor can have 4300 revolutions per minute per volt
- Motor Type: The motor type is a Brushless DC motor which provides higher efficiency, they are long-lasting and provides better performance compared to brushed motors.

2. Technical Specifications of the motor:

- kV Rating: 4300kV (kV = RPV per volt)
- Configuration of the motor: 9N12P which means the motor has 9 stator poles and 12 magnets present in it)
- The weight of the motor is approximately around 5-7 grams per motor.

- The diameter of the shaft is 1.5mm
- The mounting pattern consists of 4 x M2 screws with 9mm hole spacing provided in it.
- The dimensions of the motor are 14.3mm x 17.6mm
- The diameter of the stator is about 11mm, and the height of the stator is 4 mm.
- The maximum continuous current is 5.2 A
- The maximum power consumed is 62.4 W
- A 3-inch propeller (GEMFAN 3016-3) is used for proper balancing and sufficient flight time
- The high-quality bearings of dimension ɸ 4 x ɸ1.5 x 2mm

3. Battery Compatibility :

- The motor is compatible with 1S to 2S LiPo batteries.
- Mostly a 2S battery is recommended
- The operating voltage of the selected battery is 7.6 V(nominal)

4. Performance Data

- The performance of the motor depends on the propeller selected and the throttle levels at which the motor is run.
- At 100% throttle for a 2S battery at 7.6 V using a 3-inch propeller is about 97.533g.

5. Wiring Information

- The wires used in the motor have different color coding, the black wire used act as ground, red wire as a positive connection where power is input and the yellow wire is the signal wire.

6. Mounting of the motor

- The M2 screws are used to secure the motor to the frame. The screws used are small to prevent any further damage to the motor windings.
- The motor should be properly installed, and the propeller should be attached in the right direction (clockwise or counter-clockwise) for optimal lift.

7. Maintenance

- The motors must be kept clean and free from dust, dirt, and moisture
- The bearings must be lubricated frequently
- Make sure that the motors are provided with sufficient ventilation to avoid overheating.

8. Cautions and Safety Warnings

- Avoid running the motor at full throttle for long hours without proper ventilation to prevent the motor from getting damaged.
- Use a battery monitoring system or alarm to prevent over-discharging of the batteries
- The motor efficiency can be varied by varying the throttle at which the motor should run. The selection of the proper throttle level helps in extending the life of the battery.
- Make sure that the propellers are tightly and properly mounted to the motors to avoid vibrations that can damage the motor.

9. Troubleshooting

- In the case of motor stuttering, check if there are any loose connections or try recalibrating the ESC
- When the motor is not spinning in that case check for any loose connections, or short circuits and check for the ESC programming.
- In case of overheating, check whether the motor is running above its rated voltage.

3.3.2 Propeller

Figure 4: Gem fan 3016 Propeller

1. Overview and Specifications
    - GEMFAN 3016-3 is the model's name of the propeller used
    - A 3 blade 3-inch propeller
    - Made up of durable polycarbonate
    - Used mainly in micro drones for high efficiency and responsive control
    - The pitch size is 1.6 inches
    - The center hole inner diameter is 1.5 mm
    - The total weight of the propellers is 1.18g

2. Performance
    - The propellers help in providing a balanced thrust and efficient performance with 2S batteries.
    - Works best with 1104 4300KV motors to deliver maximum thrust.

3. Installation
    - The propeller blades should be properly mounted to ensure optimal lift.
    - There are 2 sets of clockwise propellers rated as 3016 and 2 sets of counterclockwise propellers rated as 3016R.
    - The 3016R propeller must be mounted to the motor shaft rotating in the counterclockwise direction and vice versa.

4. Safety and Maintenance

- The propellers must be durable to withstand any damage from crash landings.
- The propellers must be guarded using a propeller guard to protect the propeller and to provide safety.

4. ESP32-CAM Mount Plate Design

The section provides the detailed overview of the custom-designed mount for integrating the ESP32-CAM onto the drone, specifically for implementing line-following and obstacle detection functionality. The mount is designed to be securely attached to the drone's top plate, positioned above the battery strap, ensuring stable placement without interfering with other components. All design work and renderings were carried out using Autodesk Fusion 360 (Education License Version), ensuring precision and adaptability to the drone's structural requirements.

4.1 Drawings

The initial step in designing the mounting plate is to take the precise measurements of the drone's top plate. While the primary dimensions were based on the top plate, the length of the mounting plate is intentionally extended beyond the top plate's size to provide additional spacing for holding the antenna securely.

The ESP32-Cam used in the project is a development board equipped with header pins on the bottom side. To make connections with the flight controller using jumper wires, rectangular cutouts are made on both sides of the plate. The cutouts facilitate the insertion of the header pins, ensuring clean and secure connection.

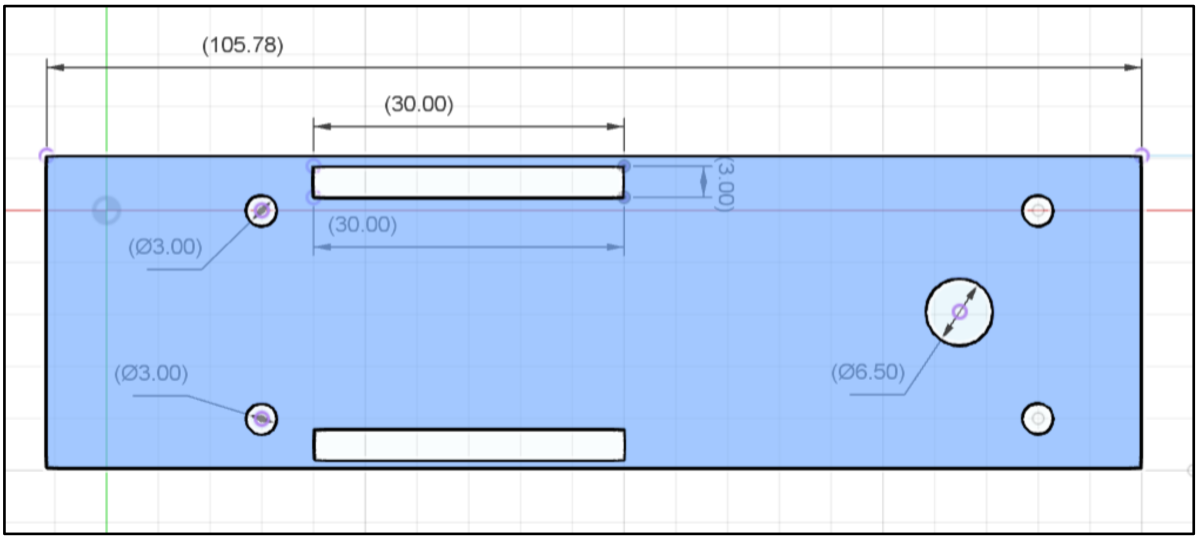


Figure 5: Mounting plate design sketch (NB: All the measurements are in 'mm')

The sketch is extruded with a thickness of 2mm providing sufficient strength without adding excess weight. Furthermore, four cylindrical columns are provided to the bottom of the mount, aligning with the screw positions on the drone's top plate. These columns allow the plate to the securely placed without impacting the drone's aerodynamics, ensuring stability during operation.

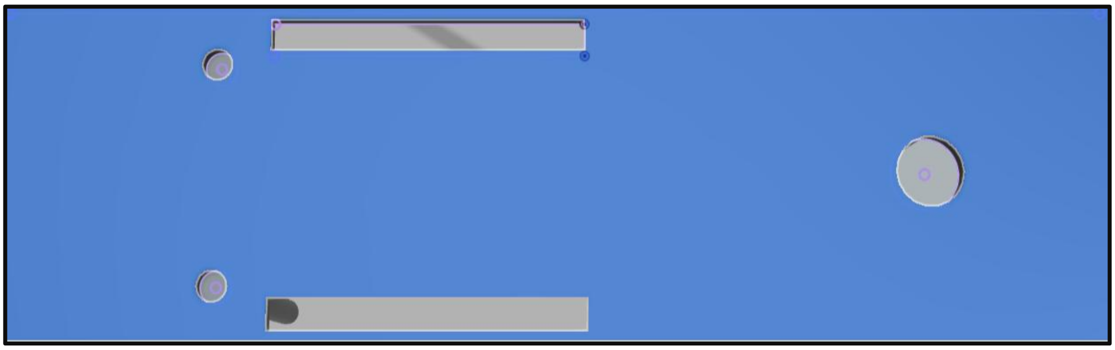


Figure 6: Final sketch with extrusion

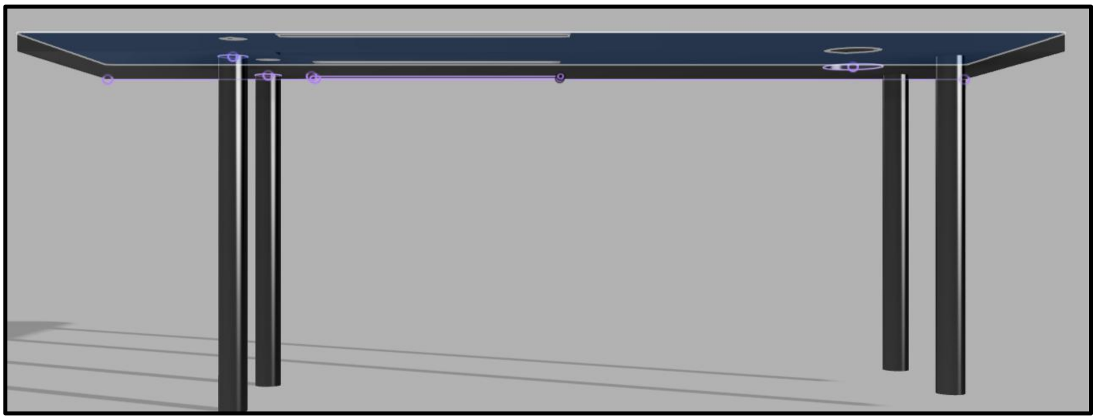


Figure 7: Rendered Model

The model file is converted into .STL format and is 3D printed after slicing it using Cura (open-source slicing software developed by Ulti maker) and converting it into G-code format. The 3D printing material used is PLA black with the print temperature around 180 degrees Celsius.

4.2 Alignment

The ESP32-Cam header pins are to be inserted through the rectangular cutout within the 3D printed model. Jumper wires are connected through the header pins to the flight controller. It is to be noted that, to tightly stick the ESP32-CAM onto the plate, adhesives such as doble-sided tape can be applied underneath the module.

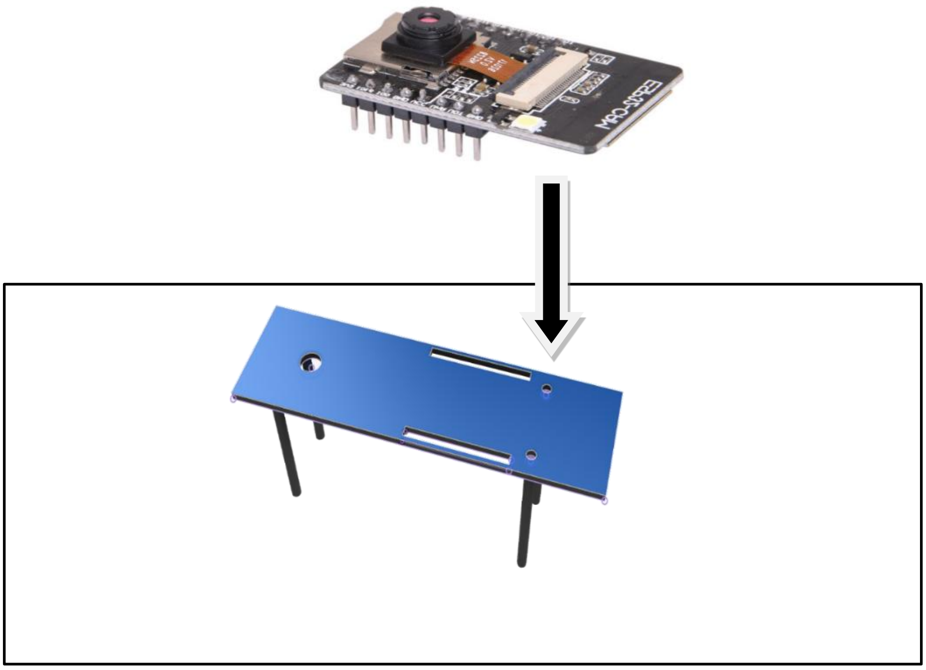


Figure 8: Insert the ESP32-CAM module to the top plate

Figure 9: Mounting plate secured on to the top plate of the drone

The final assembled module on to the top of the drone is shown in figure 5. The plate is stuck on to the top plate using hot glue gun. It is important to note that hot glue gun may not provide sufficient hold for a drone flying at high speeds and it could potentially damage the module in the event of a crash. Therefore, alternate methods should be explored for securely integrating the mounting plate to the drone, ensuring both stability and safety.

4.3 Ultrasonic Sensor Holder Design

The ultrasonic sensor mounted onto the drone using a holder obtained from an online design repository . The holder is 3D printed using the black PLA material for durability and lightweight properties. It is securely glued to the bottom position of the ESP32-CAM mounting plate ensuring a stable and effective setup for obstacle detection.

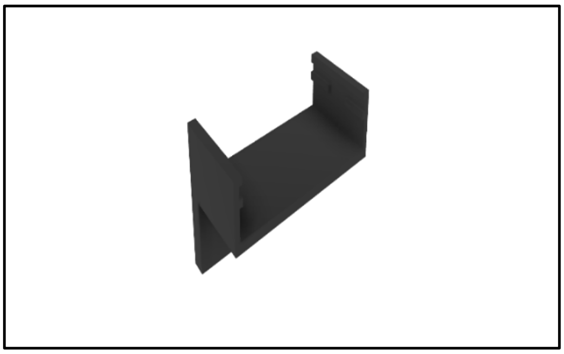


Figure 10: Mount for Ultrasonic Sensor (HC-SR04)

Figure 11: Ultrasonic sensor holder with the ESP32-Cam mounting plate

As previously indicated for ESP32-Cam mount plate, it important to note that hot glue gun may not provide sufficient hold for a drone flying at high speeds and it could potentially damage the module in the event of a crash, so care must be taken during the flight operation. For future improvements, screw holes can be provided in the holder, so that it can be screwed to the bottom plate of the drone, but this can affect the aerodynamics of the overall system which might affect the stability of the drone during flight.

4.4 Propeller Guard Design

The propeller guards are 3D printed from files obtained from an online repository []. Four propeller guards, created using PLA Black material, are affixed to the arms of the drone. Prior to installation, the printed components require some filing to ensure proper fit to the arms.

Figure 12: 3D print model for Propeller guard

However caution must be taken to avoid excessive filing as this could result in a loose fit and compromise the intended functionality of the propeller guards. To enhance the security of the installation, hot glue is applied to further secure the propeller guards to the drone.

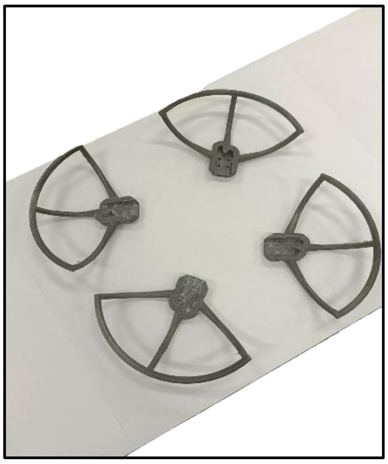


Figure 13: Printed Propeller guards

4.5 Landing Gear Design

The landing gears are 3D printed from files obtained from an online repository. To prevent the damage of critical components of the drone landing gears are crucial. The files are in 4 different parts and printed separately. Once printed, these are then attached together. The ‘V’ shaped bars are fixed to the bottom plate of the drone using some screws. This might lead to some stability issue during flight. Due to the unsuccessful flight attempts, it is unable to fully comprehend the operational dynamics of the drone. Once the issues with the flight controller commands are resolved, it will be essential to assess the stability of the drone during subsequent test flights.

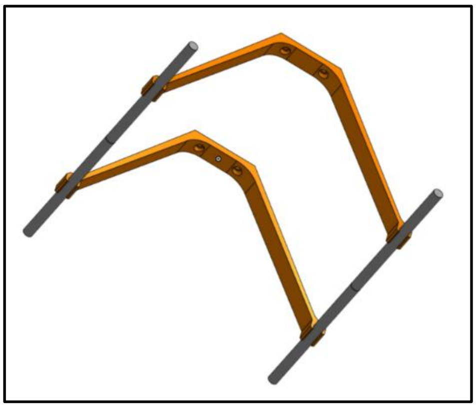


Figure 14: Landing Gear

4.6 Final Design

The final product for the end user is illustrated in the figure below. A comprehensive step-by-step guide for its operation is provided in the subsequent sections.

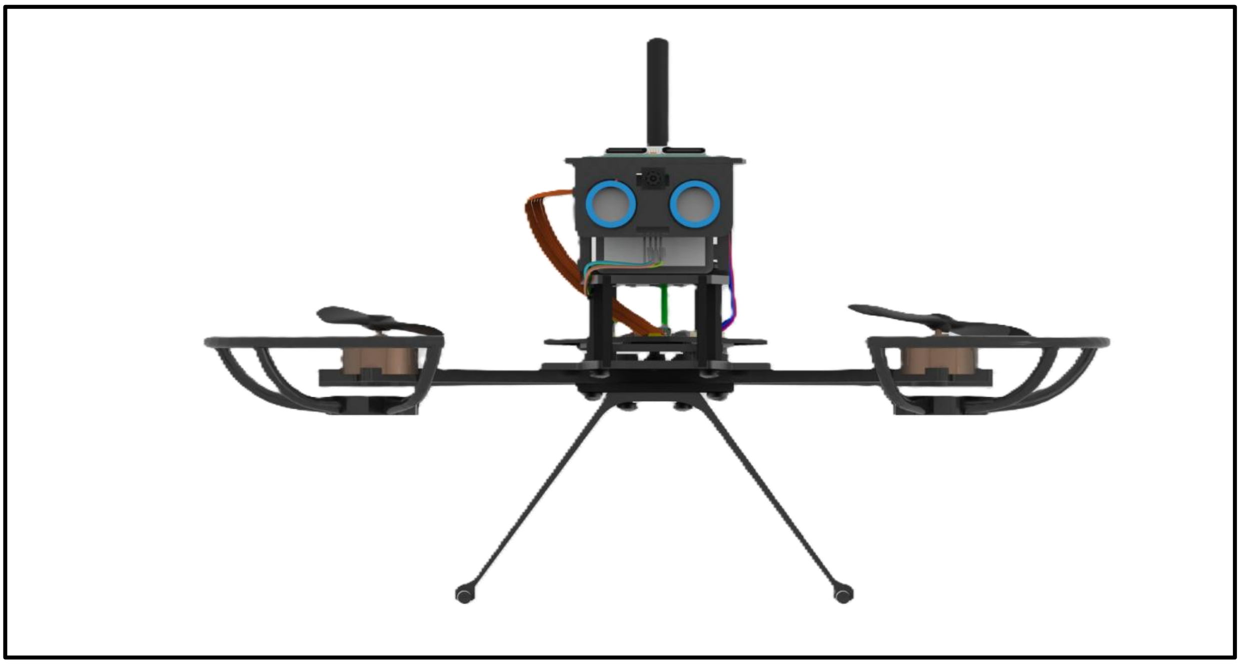


Figure 15: Drone Prototype Front View

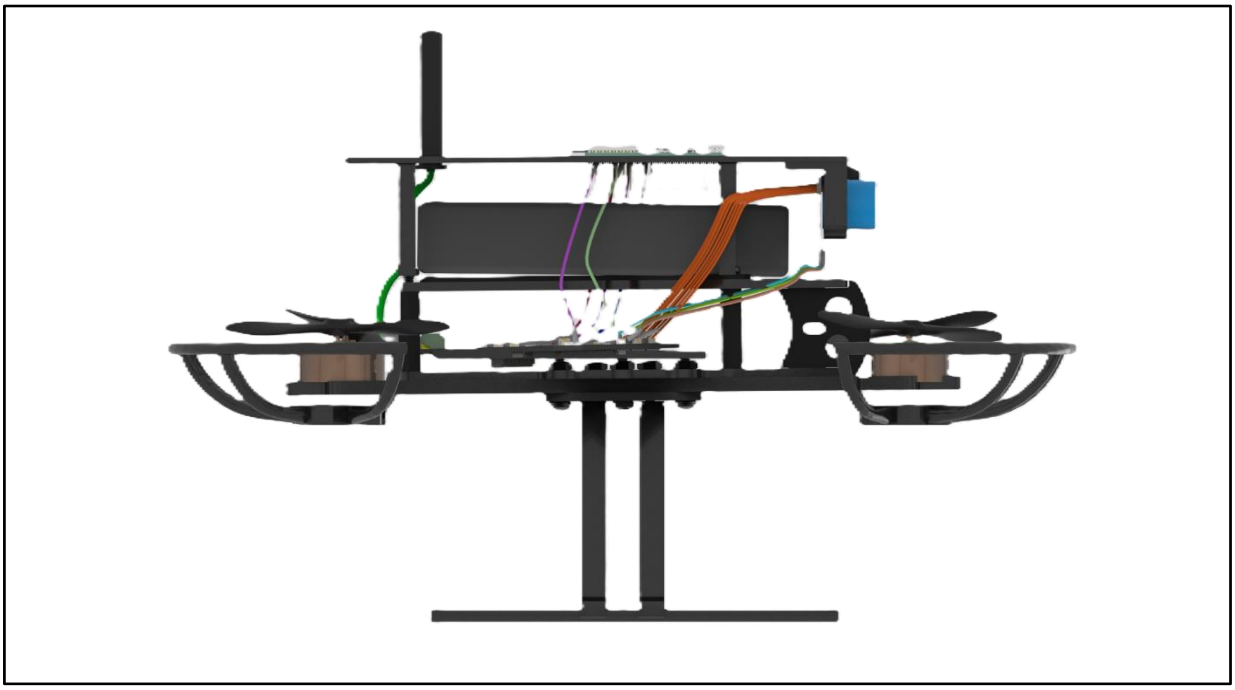


Figure 16: Side View

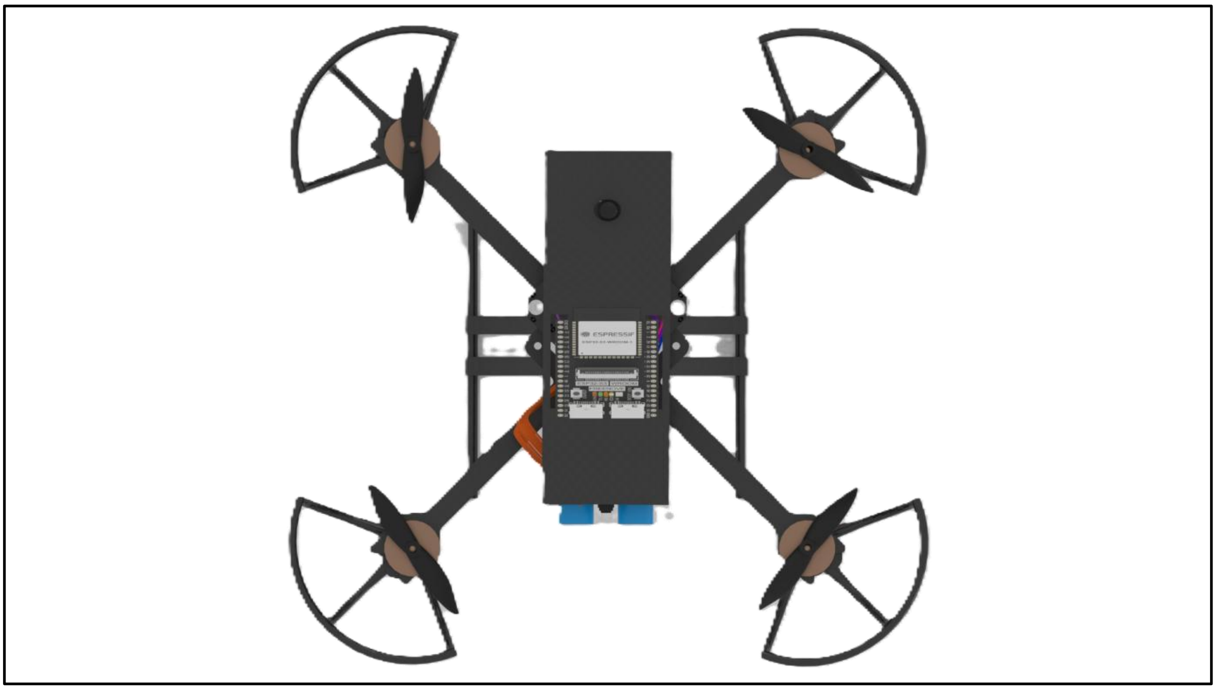


Figure 17: Top View

NB: The figure above shows 2-blade propeller; however the system uses a 3-blade propeller.

5. Part 2- Software Integration

5.1 Setting up Pycharm IDE

This section provides step-by-step instructions for installing PyCharm, setting up the necessary environment, and running the code for the drone line-following project using OpenCV and ESP32-CAM.

Step 1: Install Pycharm

Pycharm is an Integrated Development Environment (IDE) used to write and run Python code.

5.1.1 Download Pycharm:

- Go to the official PyCharm website: https://www.jetbrains.com/pycharm/download
- Download the Community version

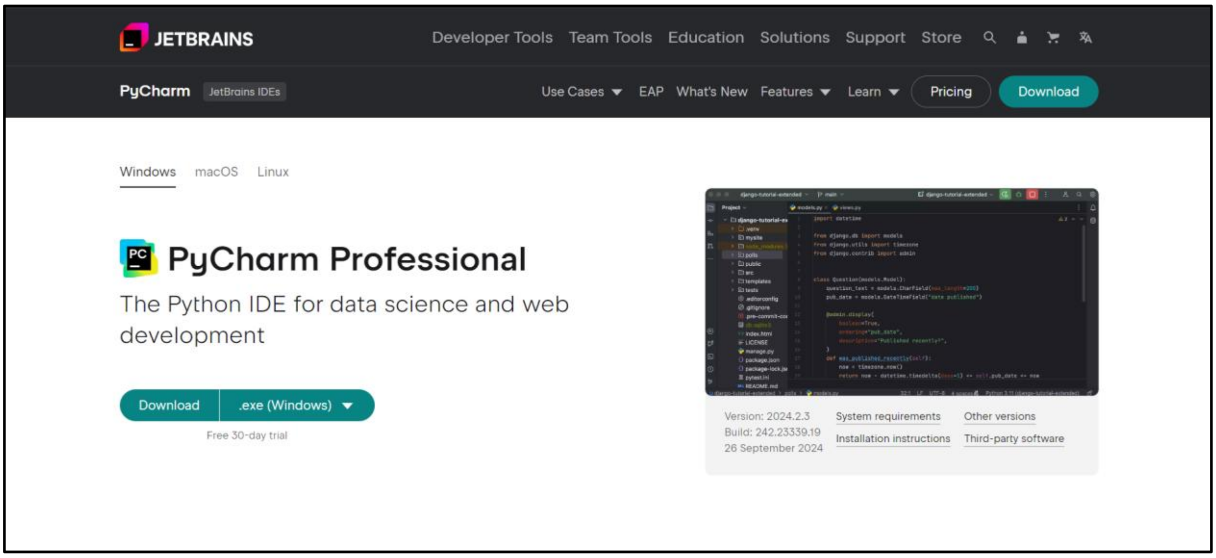


Figure 18: PyCharm download window

5.1.2 Install Pycharm

- Open the downloaded file
- Follow the installation wizard
    - Accept the terms and conditions.
    - Choose a destination folder.
    - Optional: Select to add PyCharm to your path for easier access.
    - Complete the installation.

5.1.3 Launch Pycharm:

- Once installed, open Pycharm.

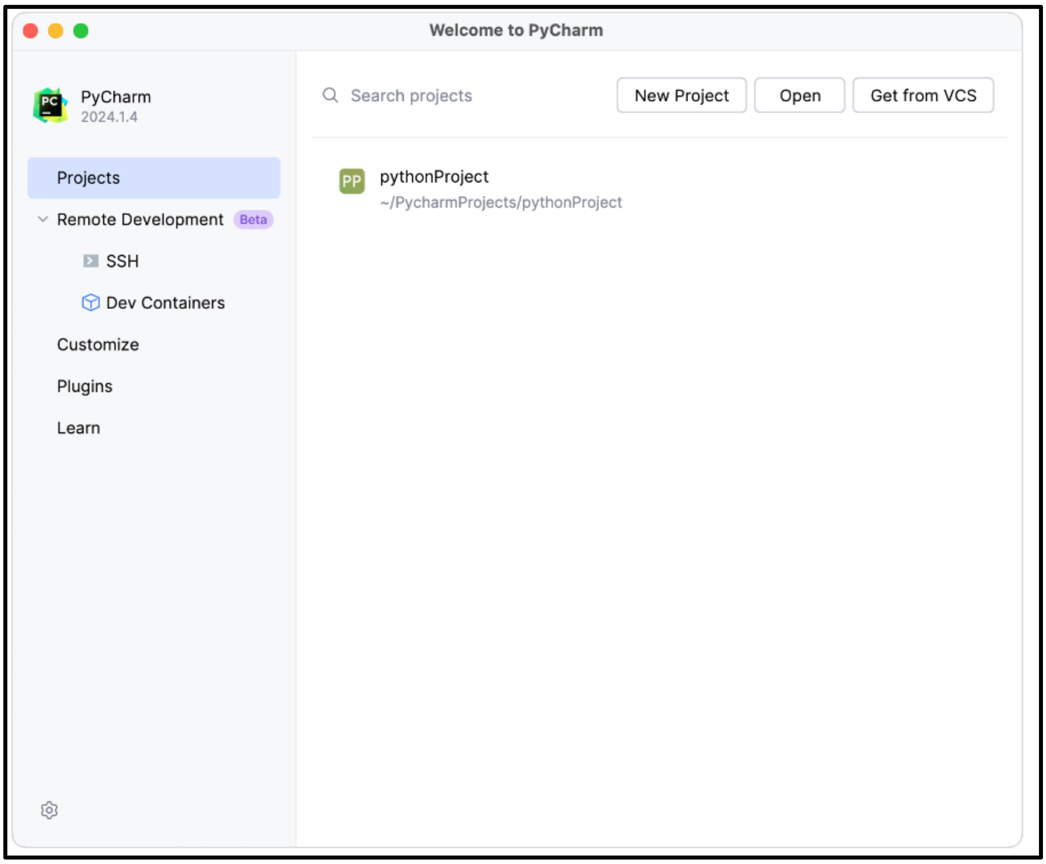


Figure 19: PyCharm welcome page

5.1.4 Download Line Following Code:

Download the linefollowing.py file from the GitHub repository: https://github.com/geobenedict/Team-18---Indoor-Line-Following-Drone or MS Teams folder where the project is shared.

- Once downloaded, select Open to load the folder containing the linefollowing.py code.

Step 2: Set Up the Project Environment

Install Python

- Download Python from: https://www.python.org/downloads/
- During installation, ensure to check the box Add Python to PATH.
- Once installed, return to PyCharm, and it should detect the Python installation automatically.

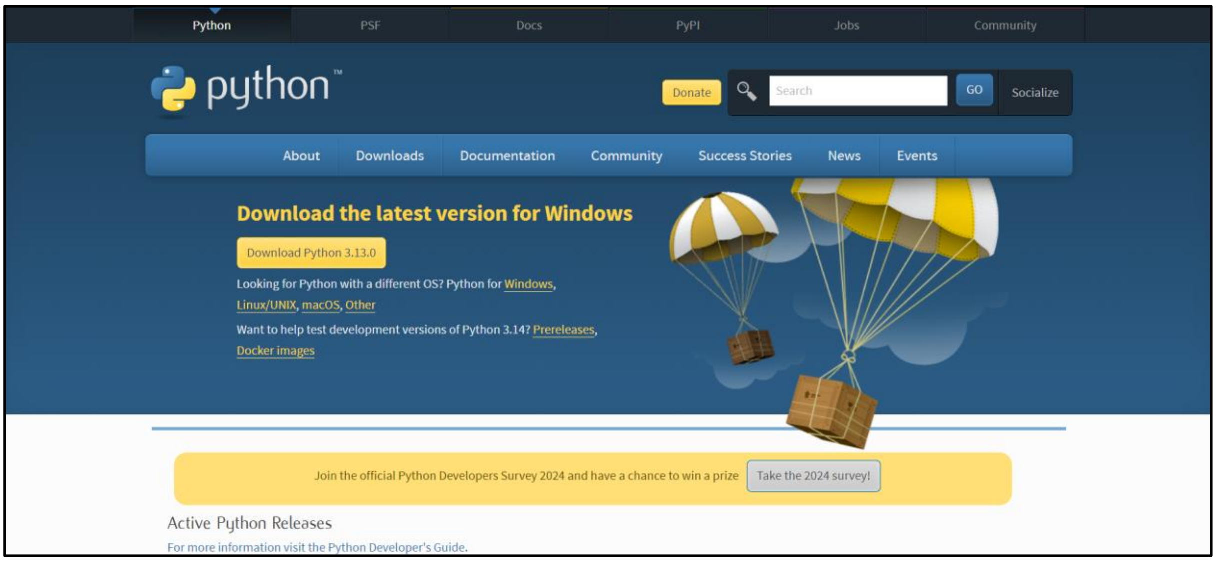


Figure 20: Python download window

Step 3: Install Required Libraries

1. Open Terminal in PyCharm: Navigate to View > Tool Windows > Terminal.

Figure 21: PyCharm Terminal

2. Install Libraries: Run the following commands in the terminal to install required libraries

   pip install opencv-python
   pip install requests
   pip install numpy

Step 4: Run the Code

1. Connect to ESP32-CAM:

Ensure that your ESP32-CAM is connected and set up as an access point, streaming video to your computer.

- Follow the instructions in the ESP32-CAM setup section

2. Configure the IP Address in the Code:
   - Inside the code, locate the line where the IP address of the ESP32-CAM is specified. It might look like:
     url = 'http://192.168.4.1/capture'
   - Ensure the IP matches the one generated by your ESP32-CAM. Modify it if needed.
3. Run the Code:
   - In PyCharm, locate the Run option at the top of the interface or right-click on the main code file and select Run.

Figure 22: Run icon in PyCharm

- The program will begin to execute, capturing the video stream from the ESP32-CAM and processing it to detect the line.

Step 5: Viewing the Results

1. Console Outputs:

- While the program is running, observe the console output in PyCharm.
- You should see information about the drone's movement commands (e.g., forward speed, rotational adjustments).

2. Graphical Output:
   - A window should open displaying the real-time video stream from the ESP32-CAM, with overlays showing detected lines and any adjustments being made to the drone's path.

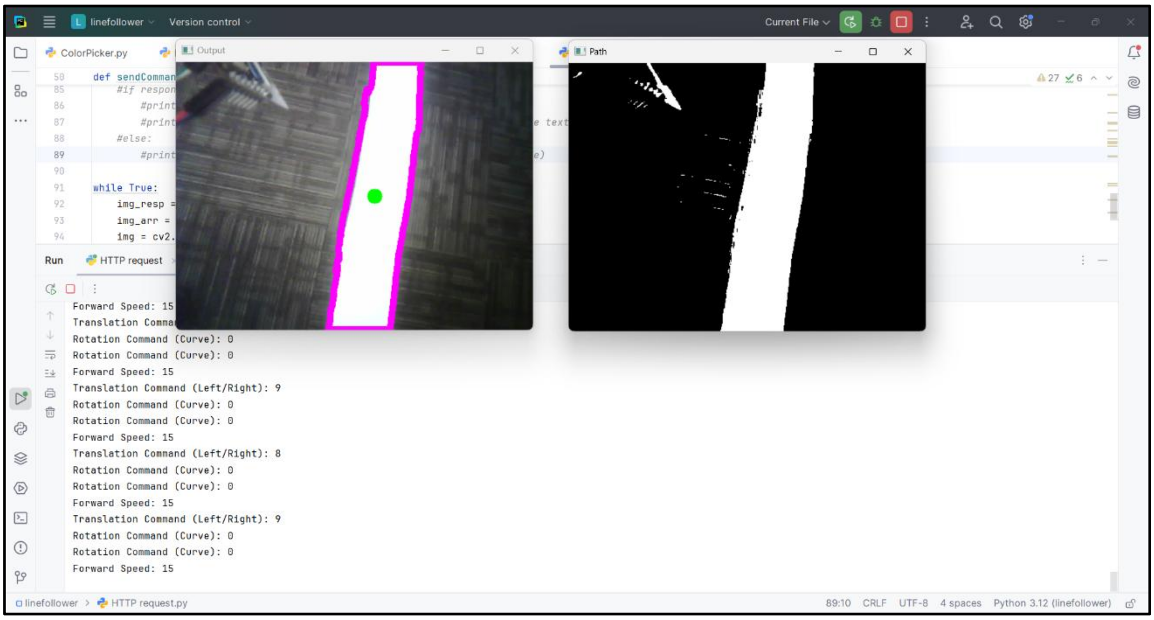


Figure 23: Output

Step 6: Troubleshooting

1. No Video Feed:
   - Ensure the ESP32-CAM is properly set up and connected to your local network.
   - Double-check the IP address in the code and confirm the ESP32-CAM is streaming.
2. Missing Libraries:
   - If you encounter missing library errors, return to Step 3 and ensure all necessary Python libraries are installed.

5.2 Ultrasonic Obstacle Detection

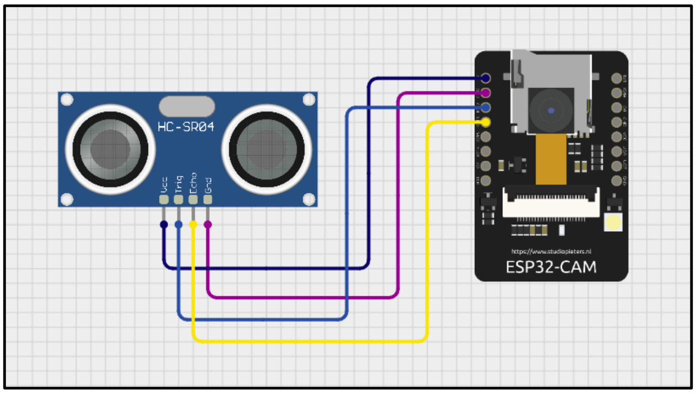


Figure 24: Ultrasonic sensor wiring diagram

Steps to view the results in the serial monitor

Setup and Connect the Hardware:

1. Ensure the ESP32 CAM and ultrasonic sensor are properly connected as per the wiring diagram.
2. Connect the ESP32 CAM to your computer via FTDI for flashing and monitoring.

Open the Arduino IDE:

1. Open the Arduino IDE on your computer.
2. Ensure the necessary board drivers for ESP32 are installed. If not, go to File > Preferences and add the following URL to the additional board manager URLs: https://dl.espressif.com/dl/package_esp32_index.json
3. Then go to Tools > Board > Board Manager, search for ESP32, and install it.

Select the Correct Board and Port:

1. Go to Tools > Board and select ESP32 CAM
2. Then go to Tools > Port and select the port where the ESP32 CAM is connected.

Upload the Code:

1. Load your obstacle detection code (the one using the ultrasonic sensor and ESP32 CAM).
2. Press the upload button to compile and upload the code to the ESP32 CAM.
3. Make sure to make the **BOOT** connection i.e, connect pin Io0 to Gnd as show in figure below on the ESP32 CAM when the IDE starts uploading the code.

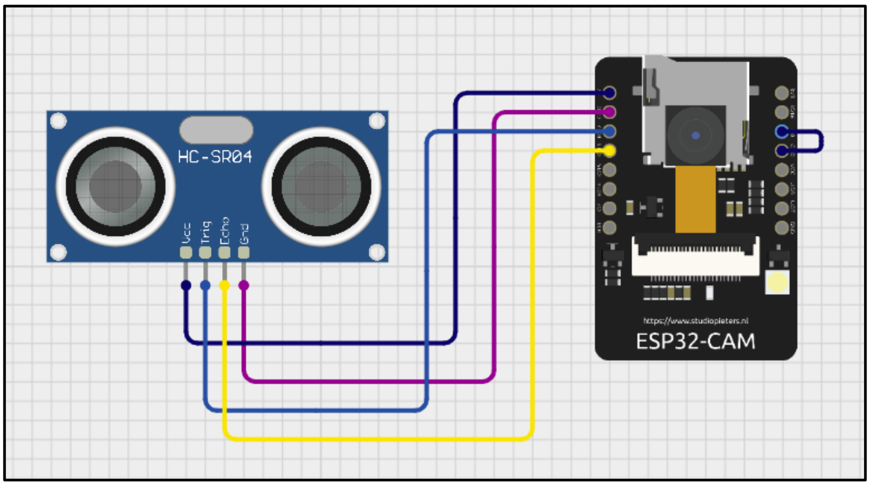


Figure 25: Ultrasonic sensor wiring diagram for uploading the code

Open the Serial Monitor:

1. Once the code is successfully uploaded, go to Tools > Serial Monitor in the Arduino IDE.
2. Set the baud rate to 115200 at the bottom of the Serial Monitor to match the baud rate in your code.

Observe the Data:

1. As the drone operates, you should see real-time data in the Serial Monitor, such as the distance measured by the ultrasonic sensor.
2. The Serial Monitor will display distance readings in centimetres and print messages when obstacles are detected (e.g., “Obstacle detected! Hovering...”).

5.3 Accessing ESP32 CAM Wi-Fi Access Point

Power on the ESP32 CAM:

1. Connect the 5v and Gnd pins of the ESP32 to appropriate power source. Alternatively, a 3.3 v power supply can also be used.

Connecting to the Wi-Fi

1. Take around 30-60 secs for the ESP32 to turn on the Wi-Fi once powered up
2. On you laptop or any device, navigate to the Wi-Fi settings and select the ESP32cam and enter the network security key = 123456789 . The SSID and security key of the WiFi Access Point can be changed by rebooting the Arduino ino file with the steps provided in step 3.

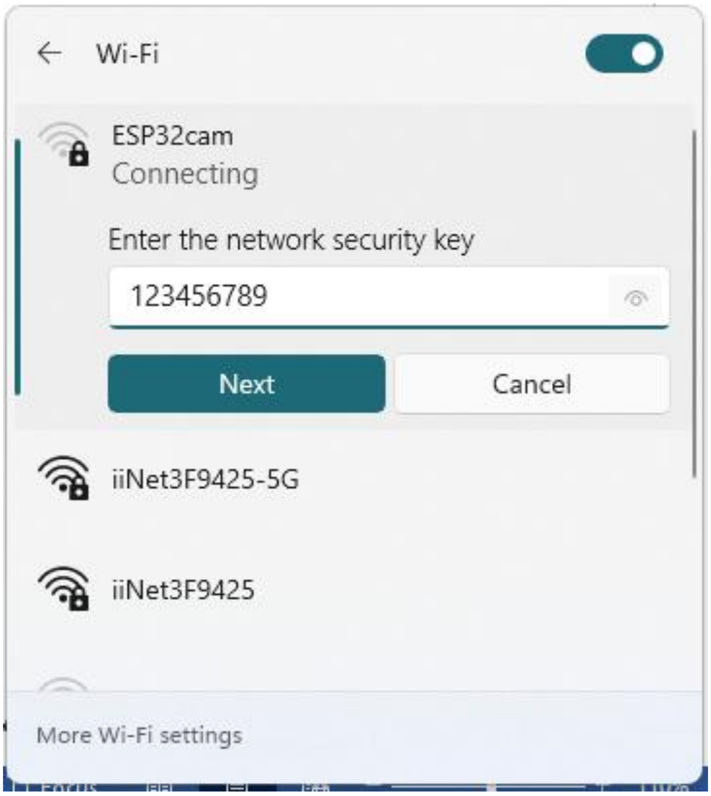


Figure 26: Wi-Fi connection using ESP32-CAM

3. To modify the SSID and password, just change the characters in green in Arduino IDE.

```
const char* ssid = "ESP32-CAM Access Point";
const char* password = "123456789";
```

Figure 27: User id and password

5.3.1 Accessing the video feed

1. Ensure the device is connected to the ESP32's Wi-Fi access point.
2. Open a web browser on the connected device.
3. Enter the default IP address: 192.168.4.1 and Enter
4. The camera interface should load as displayed below.

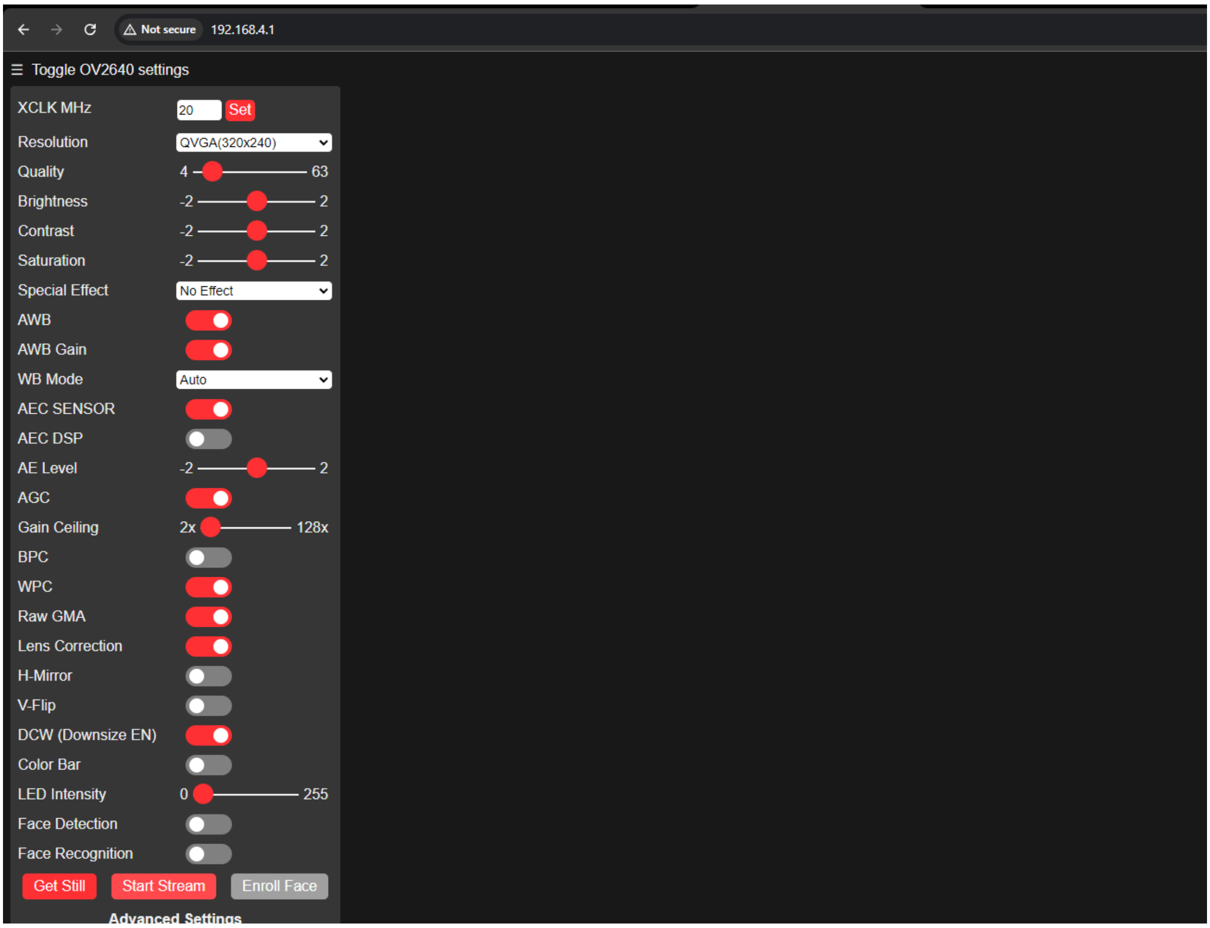


Figure 28: IP address camera feed

5. Press Start Stream to see the video feed.
6. Also, it should be noted that this IP address can be fed into PyCharm using the OpenCV library to process the feed straightaway.

5.3.2 Connecting external antenna

1. By default, the ESP 32 CAM module utilizes the built in PCB antenna.
2. To connect an external antenna, the resistor on the bottom must be soldered as shown in the figure below

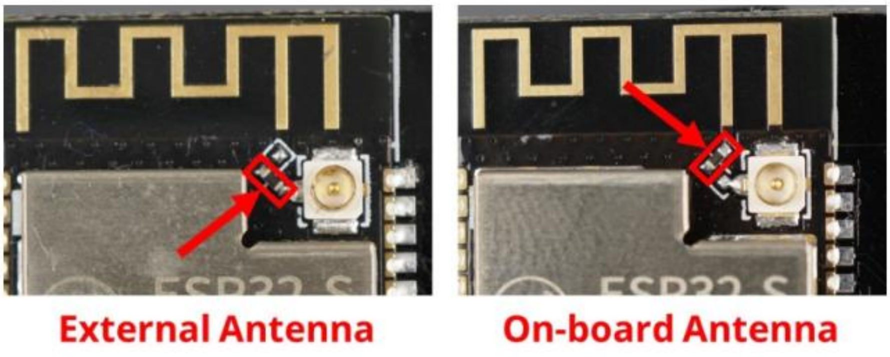


Figure 29: external and on-board antenna solder connection

3. In the ESP32 CAM module provided along with the drone is soldered for external antenna configuration. So, it can be straight away used by just connecting the UFL connector of the antenna to the ESP32 module.

6 Part 3 – Step-by-Step Instructions Guide

The section provides a step-by-step process to safely operate and fly the drone prototype. Although the designed system failed to achieve a successful flight because the flight controller did not respond to the commands sent through the UART port, the following steps provide a guide for the successful operation if the flight controller is replaced or a new 4 in ESC is purchased, as part of future improvements.

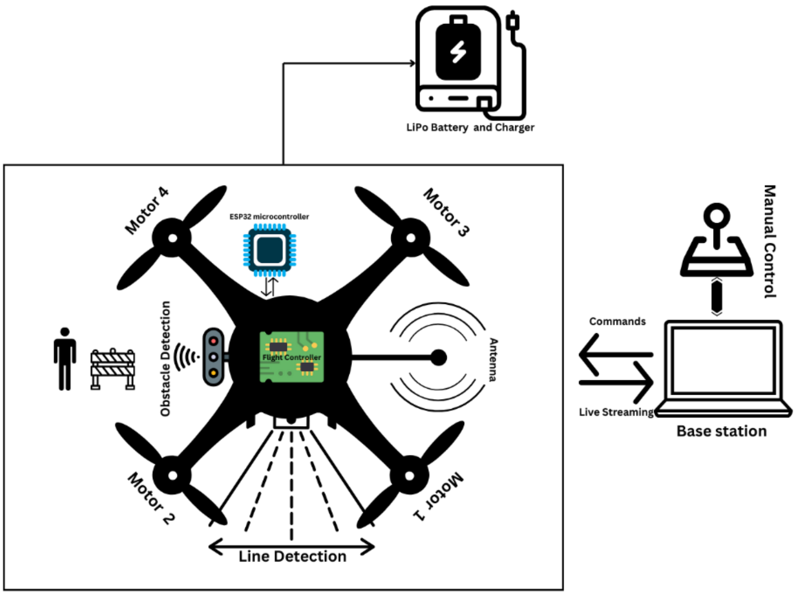


Figure 30: Basic Design Overview

Step1: Connect the drone to the battery.

- Insert the drone’s battery securely into its designated compartment.

- Ensure the battery is fully charged before use.
- Ensure the Esp32-Cam and the ultrasonic sensor is properly aligned to its respective positions and tightly fixed.

Step2: Place the drone to the Test path

- Position the drone on the floor or corridor where the test will be conducted.
- Ensure the test area is well-lit with a visible line (track) marked on the floor.
- Ensure the area is clear of obstacles for the first few meters of the test path.

Step3: Initialise the Flight

- Activate the drone's flight mode by pressing the start button. This button acts as an emergency switch for manual shutdown if needed.
- Once the start button is pressed, the drone will power up and take off automatically to a height of 1.2 meters. The drone will hover at this altitude, awaiting navigation instructions.

Step4: Run OpenCV code

- On the laptop, ensure that the OpenCV is properly set up and the line-following code is ready.
- Connect the laptop to the drone via Wi-Fi(ESP32-CAM) module.
- Start running the line following code. This step will ensure the video feed from the drone to your laptop is running for enabling OpenCV to track the line.

Step5: Obstacle Detection and Response

- If an obstacle is detected within 30cm, the drone will automatically hover in place.
- The drone will wait until the obstacle is cleared before resuming navigation.

Step6: Handle T-junctions and intersections for completion of flight

- When the drone encounters a T-junction or an intersection on the path, it will identify it as the end of the test route.
- The drone will stop at the intersection and prepare for a safe landing procedure.

Step7: Safe landing

- Once the drone identifies the T-junction, the landing sequence will begin.
- Ensure the landing area is clear of any obstacles.
- The landing gear helps to achieve a smooth landing.

Step8: Emergency Switch

- In case of communication failure, press the emergency switch for turning to manual control. Here, in the designed prototype, emergency switch is not implemented, and this might lead to control failure during emergency situations.

Step9: Flight Shutdown

- Once the flight is complete, disconnect the battery from the drone to power it down.

7. Conclusion

The comprehensive user guide and manual presented above provides a detailed overview of different components, assembly process and steps required for achieving a successful flight. Each section elaborates on the individual parts of the drone, including mounting peripherals, preparing the system and ensuring proper functionality. However, due to the system integration challenges experienced during the testing phase, it is recommended to consider using a different flight controller or ESC in future iterations to successfully demonstrate the line following functionality. This guide serves as a foundation for further improvements and development of the drone system.