

AUTONOMOUS FLIGHT SETUP

1

Overview: This document outlines the complete workflow to set up and run an autonomous drone mission using Pixhawk (ArduPilot), Raspberry Pi, a YOLOv8-based litter detection system, and a servo picker mechanism. It includes configuration steps, code placement, and system architecture.

I) ArduPilot Configuration (Using Mission Planner on PC) Software: Mission Planner (Windows)

Steps:

1. Connect the Pixhawk via USB.
2. Flash ArduCopter firmware from the "Install Firmware" tab.
3. Calibrate: - Accelerometer - Compass - Radio (RC) - ESCs
4. Connect: - GPS - Telemetry Radio - Power Module
5. Configure Flight Modes: - Stabilize / AltHold (for testing) - Loiter (for hovering during detection) - AUTO (for autonomous missions) - GUIDED (for dynamic control via Raspberry Pi)
6. Set Return to Launch (RTL) as the failsafe action.
7. Upload Patrol Waypoints using the waypoints.txt file via PLAN > Load WP > Write WPs.

II). Autonomous Mission Script (mission.py) Location: Raspberry Pi

Steps:

1. Transfer mission.py to the Raspberry Pi.
2. Ensure DroneKit is installed.
3. Connect Pixhawk via serial (/dev/serial0) or USB.
4. Run the script. Purpose: Manages arming, takeoff, patrol, detection pause, litter pick-up, and return to launch (RTL).

III) YOLOv8 Litter Detection Script (yolo_detect.py) Location: Raspberry Pi

Steps:

1. Install dependencies.
2. Edit mission.py to import the detection function.
3. Implement is_litter_detected() in yolo_detect.py to return True when litter is detected with high confidence.

AUTONOMOUS FLIGHT SETUP

2

IV)Ground Station Dashboard (Flask Web App) Files: app.py (backend), dashboard.html (frontend) Location: Raspberry Pi

Steps:

1. Place both files in the same directory. Autonomous Drone System with ArduPilot, Raspberry Pi, and YOLOv8 Integration
2. Install Flask.
3. Run the dashboard.
4. Access in browser via `http://:5000/` Function: Displays live telemetry (GPS, battery, altitude, and flight mode)

V)Servo Picker Configuration

GPIO Pin Mapping (BCM): - 17: Arm movement - 18: Claw open/close - 27: Optional (e.g., wrist rotation) Wiring Notes: - Connect servos through a servo driver (e.g., PCA9685) or directly via GPIO with external power. - Tune angles in `pick_litter()` according to your mechanical design.

VI)Folder & File Structure /home/pi/drone_project/ mission.py yolo_detect.py /home/pi/drone_dashboard/ app.py dashboard.html

- Replace the dummy `detect_litter()` logic with live YOLOv8 detection.
- Adjust servo angles to match your arm/gripper mechanics.
- Test in STABILIZE mode before switching to AUTO. - Optionally, configure the system to auto-launch on boot using `systemd` or `cron`.

Prepared by : NIZAM PM, ATHULKRISHNA NS, SRAVAN CN, NIDHIN BABY

Date : 09/04/2025

Project : Autonomous Litter Detection And Picking Drone