Technical MANUAL

Line-Following Drone

Team01

October 2024

USER'S MANUAL

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

A. GENERAL INFORMATION

1.1 System Precautions

The safe operation of this drone relies on mitigating hazards to the surrounding environment, personnel and the product itself prior to operation. To achieve this, please adhere to the following recommendations:

1.1.1 Propellor Mounts

On top of each propellor, there are two Phillips-head screws that mount the blade to its relevant brushless motor. By applying torque with a standard Phillips screwdriver, ensure that each propellor is securely mounted.

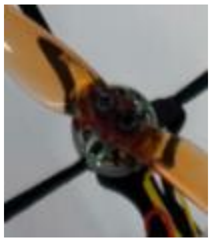


Figure 1 Propellor Screw Positions

1.1.2 Intrusive Cabling

During manual handling, it is possible for the peripheral sensor or motor cables to protrude from their zip ties and make their way to the propellor region. Ensure that all these cables are moved away from the blades and secured in their zip ties.

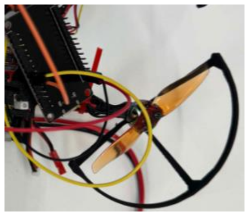


Figure 2 Example of Intrusive Cabling

1.1.3 Safe Environment

As this drone is intended for indoor usage, alterations must be made such as enforcing the use of PPE (such as glasses) and keeping fire-extinguishers on hand as opposed to free flight in open areas. It is recommended the drone flies on a flat surface with a predefined line-marker.

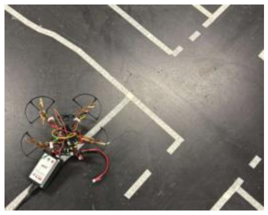


Figure 3 Example of flat surface required for testing

1.2 System Prerequisites

The following materials are required for successful manual and autonomous operation of the drone.

- 1x GPIO for connecting LiPo wall-charger.
- 1x Laptop with integrated Wi-Fi capabilities.

The following software is required:
- Operator GUI application provided in this suite.
- Python3 installed on the operator's laptop with the following external packages.
  - TKinter
  - OpenCV
  - Sockets

These tools are not required but significantly ease diagnostic processes:
- 1x 0x50mm Phillips-head Screwdriver
- 1x M2 Allen key
- 1x Spare 2S-800mAH battery.

1.2 Acknowledgement of Limitations

Despite this product being intended for indoor monitoring of MRG's resources, it is important to acknowledge technical limitations that are involved.

An onboard monitoring system ensures that the drone hovers on the spot in the presence of an obstacle. Whilst this is true, the system is significantly limited in the range and distance it can observe obstacles from. Although a hard limit of 20cm is set for the obstacle, electronic noise in the sensor readings may require user tuning of inputs to account for *actual* distance.

In a similar manner, the height stabilisation sensors are comprised of ultrasonic sensors. As a result, the same uncertainty or noise will be present. Manual tuning will be required by the operator in order to effectively operate.

1.3 Preflight Checks

1. **Power On and Connectivity**:
    - Ensure the drone's battery is fully charged.
    - Power on the drone and confirm that all onboard systems (ESP32, ESP32-CAM, flight controller) are functioning. This can be done by checking whether a RED LED is on for each component.
    - Connect the controller system to the drone via Wi-Fi (SSID: "ESP32-Test-AP", Password: "12345678").
    - Confirm that the IP addresses are correctly configured (ESP32 IP: 172.20.10.13, ESP32-CAM IP: 172.20.10.12).
    - Verify the TCP connection to the ESP32 (port 12345) and ensure the control interface shows "Connected."
2. **Sensor and Obstacle Detection**:
    - Verify obstacle detection data is being received from the ESP32 (obstacle distance shown in the GUI).
    - Ensure the distance label is updated with real-time data, indicating proper sensor functionality.
3. **Video Feed Check**:
    - Confirm that the live camera feed from the ESP32-CAM is visible in the GUI.
    - Check that the camera feed updates smoothly, and that line angle detection is operational (angle displayed correctly).
    - Verify that the image resolution and frame rate are acceptable.
4. **Control Interface Check**:
    - Test all control modes (Emergency Stop, Hover Mode, Manual Mode, Autonomous Control).
    - Send a command to the drone (e.g., HOVER, MANUAL, AUTO) and verify that the drone responds appropriately.
    - Ensure the drone state is correctly displayed in the GUI (e.g., "Hovering," "Following Line").
5. **Throttle and Flight Controller**:
    - Ensure that the throttle, pitch, and yaw commands can be sent and acknowledged.
6. **Simulation Mode (Optional)**:
    - Test the simulation by toggling it on and off to confirm drone states (e.g., "Lifting," "Hovering") update in the GUI.
7. **Failsafe and Emergency Stop**:
    - Test the Emergency Stop function and confirm the motors stop immediately.
    - Verify that a disconnection is properly handled, and the system attempts to reconnect automatically.

Once all these checks are confirmed, the drone is ready for flight.

2.0 SYSTEM OPERATION

B. SYSTEM SUMMARY

2.1 Powering on Device

1. After ensuring PPE is appropriately adorned, place the drone on a thin black line to start.
2. Whilst keeping the drone flat on the ground, plug in the LiPo XT30 connector from the battery to the flight controller. Ensure that there is no debris lodged between the connectors.
3. Listen for a dual-tone beep. Once this beep is observed it means that the system is armed and the Team01 drone is energized.

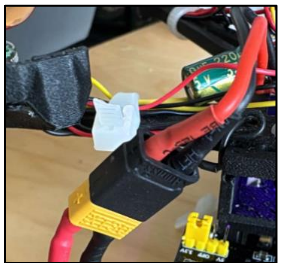


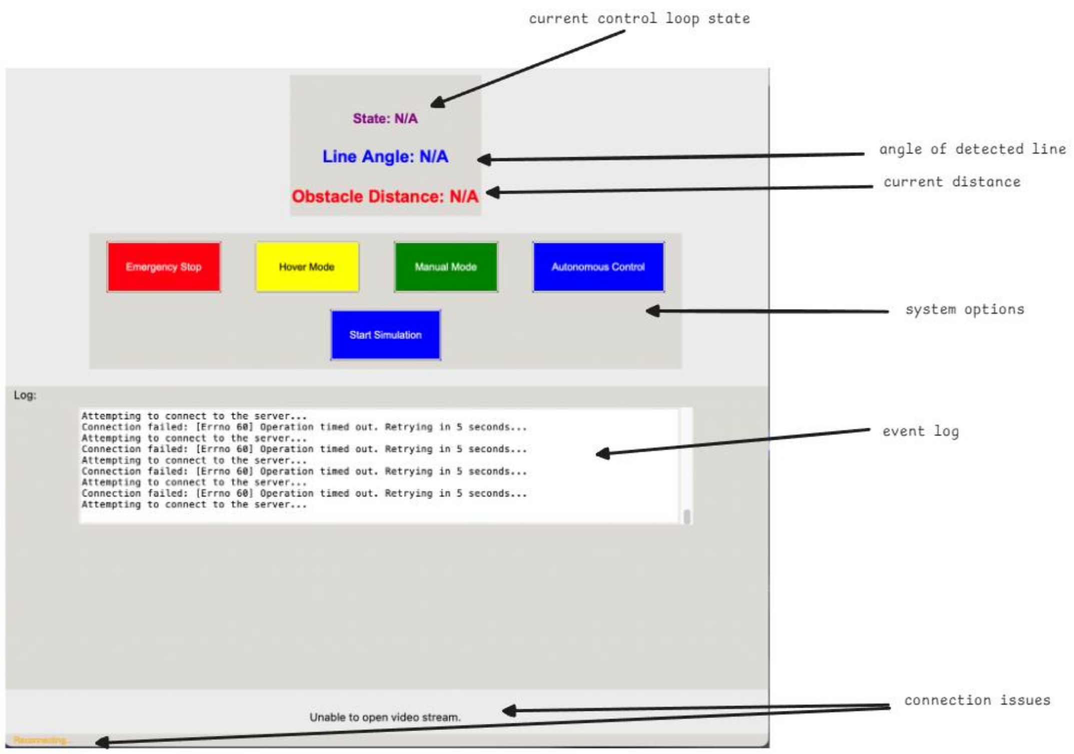
Figure 4 Initial Power Connection

4. The device is now ready for remote connection.

2.2 Connecting Controller

1. Open the Wi-Fi connection settings on the operator's device. Note that Linux, Windows and MACOS are supported.
2. Once in the network choices panel, select the **Team01 Drone** option and enter the password **12345678**.
3. After connecting and ensuring a stable network connection, launch the operator_gui.py by running the following command in a terminal window:

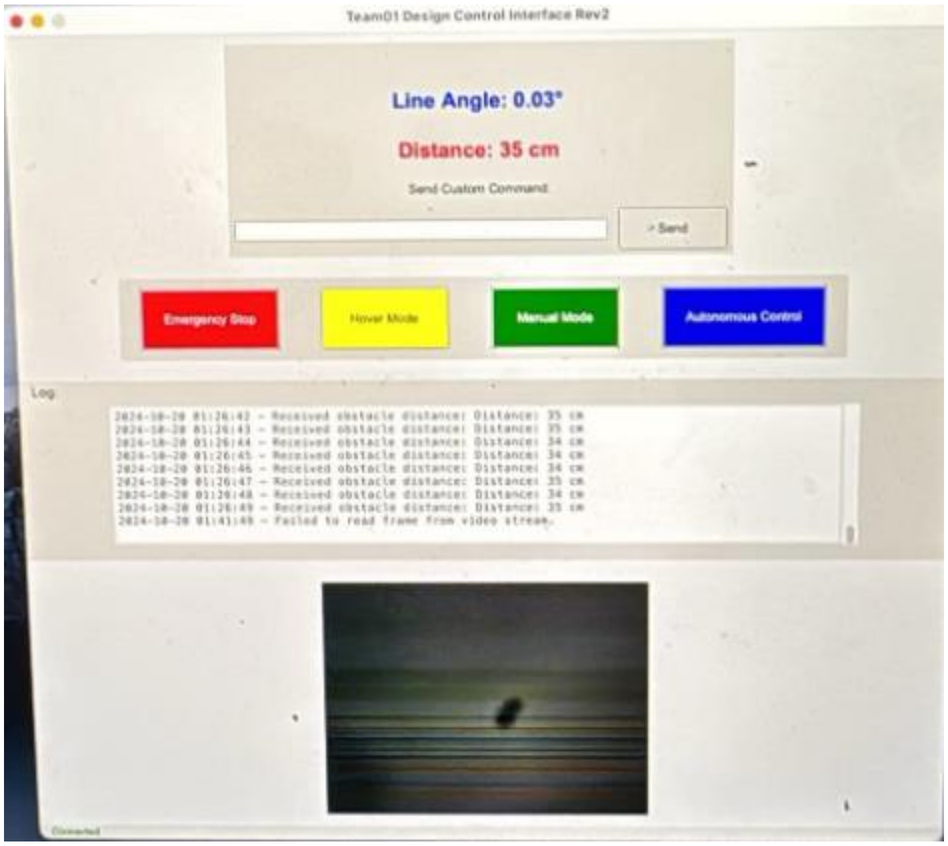
   Python3 operator_gui.py
4. Soon, a screen will launch with the following interface.



```
Attempting to connect to the server...
Connection failed: [Errno 60] Operation timed out. Retrying in 5 seconds...
Attempting to connect to the server...
Connection failed: [Errno 60] Operation timed out. Retrying in 5 seconds...
Attempting to connect to the server...
Connection failed: [Errno 60] Operation timed out. Retrying in 5 seconds...
Attempting to connect to the server...
Connection failed: [Errno 60] Operation timed out. Retrying in 5 seconds...
Attempting to connect to the server...
```

Figure 5 Operator GUI in connecting state

5. Observing the event log in the GUI, we can see that the telemetry data is slowly loading in.
6. Once fully connected, the operator should be able to see the following screen.



```
2024-10-20 01:26:42 - Received obstacle distance: Distance: 35 cm
2024-10-20 01:26:43 - Received obstacle distance: Distance: 35 cm
2024-10-20 01:26:44 - Received obstacle distance: Distance: 34 cm
2024-10-20 01:26:45 - Received obstacle distance: Distance: 34 cm
2024-10-20 01:26:46 - Received obstacle distance: Distance: 34 cm
2024-10-20 01:26:47 - Received obstacle distance: Distance: 35 cm
2024-10-20 01:26:48 - Received obstacle distance: Distance: 34 cm
2024-10-20 01:26:49 - Received obstacle distance: Distance: 35 cm
2024-10-20 01:41:49 - Failed to read frame from video stream.
```

Figure 6 Primed operator terminal

7. Ensure that the camera feed is clean by observing any debris on the feed. In the case for the image above, it is evident that a crack in the feed will produce undesired results and will require replacement.
8. Check that the line angle and obstacle distance sensor are calibrated correctly. If they are not please refer to the tuning section within this report.
9. Once steps (7) and (8) are validated proceed to the next section.

2.3 Toggling Modes

The Team01 Drone features a multitude of potential actions. Please observe the descriptions provided below and select accordingly.

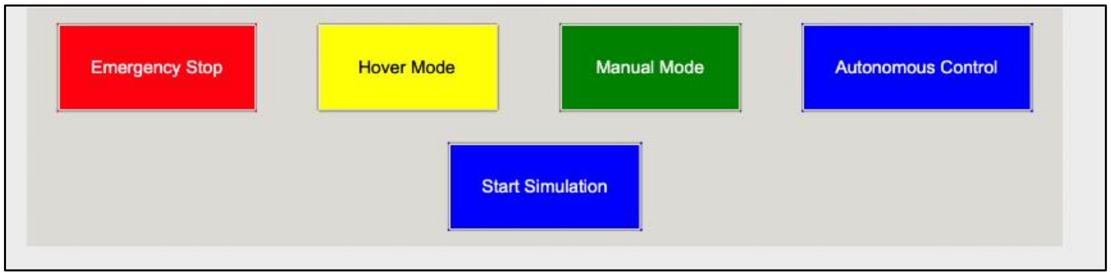


Figure 7 Example modes

Table 1: Buttons and Actions

Command	**Action**
Emergency Stop	Immediately cuts throttle to the ESCS. Note that it does not deenergise the system and will result in a rapid descent from any altitude.
Hover Mode	Will maintain flight at the given altitude, pausing continuous operation until advised otherwise.
Manual Mode	Will begin manual mode which will be later added into the system enabling settings such as manual flying forward and backwards.
Autonomous Control	Will begin the autonomous flight control system: 1. Hover to predefined height. 2. Check for obstacles (hover until obstacle removed) 3. Move to intended flight angle. 4. Continue until line angle is no longer present.
Start Simulation	Simulate the procedure outlined above.

2.4 Tuning Device

To account for dynamic environments whilst promoting user configurability, the following parameters can be defined by the operator.

Table 2: Tunable configurations

Parameter	**Definition**
Obstacle Distance	Value in centimetres.
Height Stabilisation	Value in centimetres.
Heartbeat Freq	Value for checks per minute.

These settings can be configured in lines 10-13 in the operator_gui.py file.