

DDBOAT Control Project

This project controls an Unmanned Surface Vehicle (USV), the DDBOAT, using various sensors such as GPS, IMU (magnetometer, accelerometer, gyroscope), and motor controllers. The boat is programmed to execute predefined missions including heading control, waypoint navigation, and swarm behaviors.

Note: This project is a more advanced and accomplished iteration of our previous work available at this link.

For a demonstration and explanation, please view our video here.

Table of Contents

- Requirements
- Connection Setup
- Programs Overview
 - Utilities
 - Control Functions
 - Missions
- Authors

Requirements

To run this project, you need:

- A DDBOAT equipped with GPS, IMU (magnetometer, accelerometer, gyroscope), and motors controlled via Arduino.
- Access to the DARTAP WiFi network.
- SSH access to the DDBOAT.

Connection Setup

Before running any program, follow these steps:

1. **Connect to DARTAP WiFi:**
 - Ensure your machine is connected to the DARTAP WiFi network.
2. **SSH into the DDBOAT:**
 - Open a terminal and use the following SSH command:
`ssh ue32@172.20.25.2XX`
 - Replace XX with the correct DDBOAT number.

Programs Overview

This project is organized into three main groups: utilities, control functions, and missions. While the utilities and control functions are not directly part of the high-level control system, they provide essential support for the missions.

Utilities

These programs offer foundational support and auxiliary functionalities:

- **client_server.py**
A third-party program designed to enable communication between DDBOATs.
- **write_log.py**
Implements a class for writing logs and handling print statements.
- **mini_roblib.py**
Contains mathematical functions to support navigation and control.

Control Functions

These scripts handle the boat's sensor data and core control computations:

- **calibration.py**
Calibrates the magnetometer to ensure accurate heading measurements.
- **get_gps.py**
Converts raw GPS data into a format that can be effectively exploited for navigation.
- **get_heading.py**
Utilizes the calibration data to compute the boat's current heading.
- **ws3k2_drivers.py**
Contains the WS3K2 class for controlling the boat's motors, GPS navigation, and heading

Missions

The following missions, which build upon the control functions, are organized in chronological order:

- **ms_round_trip.py**
Created to test the heading calibration. The boat performs a round-trip maneuver, initially moving in one direction and then reversing after a calibration test.
- **ms_come_back.py**
Commands the boat to navigate back to the pontoon using GPS data.
- **ms_fix_circle.py**
Directs the boat to follow a circular trajectory around a static buoy.
- **ms_circle.py**
Guides the boat along a circular path around a moving buoy (for example, another DDBOAT).

- **ms_follow_boat.py**
Enables the boat to follow another DDBOAT.
- **ms_nav_towp_swarm.py**
A swarm test mission that launches every available DDBOAT toward the same GPS waypoint.
- **consensus.py**
The most advanced mission where one boat is declared the leader to perform a specified mission, while the other DDBOATs follow its lead.

Note: You can find videos representing some of the missions directly on the README.md.

Authors

This project was developed by:

- **BOURIN Toméo**
- **DUNOT Clément**
- **FLEURY Vianney**

From **ENSTA - Autonomous Robotic 2026**.