Alma Mater Studiorum · University of Bologna

Bachelor's Degree in Computer Engineering and Science
A.A. 2025/26

Smart Tank Monitoring System

Grazia Bochdanovits de Kavna
matr. 0001117082

Alessandro Rebosio
matr. 0001130557

Contents

Chapter 1

Analysis

This chapter analyses the functional behaviour and hardware structure of the Smart Tank Monitoring System, outlining its operating sequences, safety logic, and implementation requirements. The objective is to define how the system reacts to flight commands and environmental conditions, and how each hardware element contributes to its autonomous operation.

1.1 Description and Requirements

1.2 Summary

Chapter 2

Architecture

Chapter 3

Arduino Workflow

3.1 Overview

3.2 System Dynamics

3.3 Coordination of Operations

3.4 Supporting Tasks and Context Interaction

3.5 Global Workflow

Appendix A

User Interface

Appendix B

User Guide

B.1 Setting up the Python environment

Navigate to the control unit subsystem directory and create a Python virtual environment:

```
> cd control-unit-subsystem
> python3 -m venv venv
```

Activate the virtual environment:

- On macOS/Linux:

  ```
  > source venv/bin/activate
  ```

- On Windows:

  ```
  > venv\Scripts\activate
  ```

Install the required dependencies:

```
> pip install -r requirements.txt
```

B.2 Flashing the embedded devices

The system requires flashing firmware to both the ESP32 and Arduino boards.

B.2.1 Tank Monitoring Subsystem (ESP32)

Navigate to the tank monitoring subsystem and upload the firmware:

```
> cd ../tank-monitoring-subsystem
> platformio run --target upload
```

B.2.2 Water Channel Subsystem (Arduino)

Navigate to the water channel subsystem and upload the firmware:

```
> cd ../water-channel-subsystem
> platformio run --target upload
```

B.3 Running the project

After setting up the Python environment and flashing the embedded devices, you can start the Smart Tank Monitoring System.

B.3.1 Starting the Control Unit

From the control-unit-subsystem directory (with the virtual environment activated), run:

```
> python control-unit-subsystem/main.py
```

B.3.2 Starting the Dashboard

Open the dashboard in your web browser:

```
> open ../dashboard-subsystem/index.html
```

Or navigate to the file location and open `index.html` in your preferred web browser.