Assignment nr.2
FSMs and Task-based Architectures: Smart Waste Disposal System

Luca Casadei - 0001069237
Francesco Pazzaglia - 0001077423
Mattia Morri - 0001082027

Last modified: December 2, 2024

Contents

1 Introduction

This project implements a Smart Waste Disposal System for liquid waste using an Arduino-based embedded system and a PC application for monitoring and operator control. It is based on a finite state machine (FSM) and task-based architecture.

2 Finite State Machine and Task Descriptions

The Arduino project developed through PlatformIO has the objective to manage the whole system through a change of states. In particular, we planned to divide the solution into 3 main tasks, which are:

1. **WasteTask**
   Manages waste disposal operations, the main task for the proper functioning of the entire system. It provides for the following states:
   - **CLOSED**: the neutral state of the system.
   - **OPEN**: when a user wants to throw something away he presses the open button and opens the door.
   - **RECEIVED**: reception of waste.
   - **ERRORED**: if the temperature increases too much
   - **FULL**: when the container fills up, so capacity goes to zero.
   - **EMPTYING**: when the operator empties the container.
2. **SleepingTask**
   Manages the shutdown of the system when no user is detected in the proximity of the waste container. It provides for the following states:
   - **AWAKE**: the system is in normal mode of operation.
   - **SLEEPING**: if no users were found nearby.
3. **TemperatureTask**
   Manages the temperature changes of the waste inside the container. It provides for the following states:
   - **STABLE**: the temperature is below the maximum threshold.
   - **UNSTABLE**: the temperature is above the maximum threshold for a temporarily period.
   - **DANGER**: the temperature has been above the threshold for too long.

3 Dashboard for Monitoring and Control

The Operator Dashboard, developed in Java, provides real-time monitoring and operator control for the system. The GUI interacts with the Arduino system via a serial connection, displaying the container's current status and allowing user actions.

3.1 Features

- **Visual Representation of Waste Levels**: The GUI includes a container panel, which dynamically updates to show the percentage of waste in the container.
- **Real-time Updates**: Serial communication allows the GUI to receive data from the Arduino system, such as waste level and temperature status.
- **User Interaction**: Buttons are provided to control the container, including:
  - **Empty the Container**: Decreases the waste level progressively when pressed.
  - **Restore the Container**: Resets the container waste level.