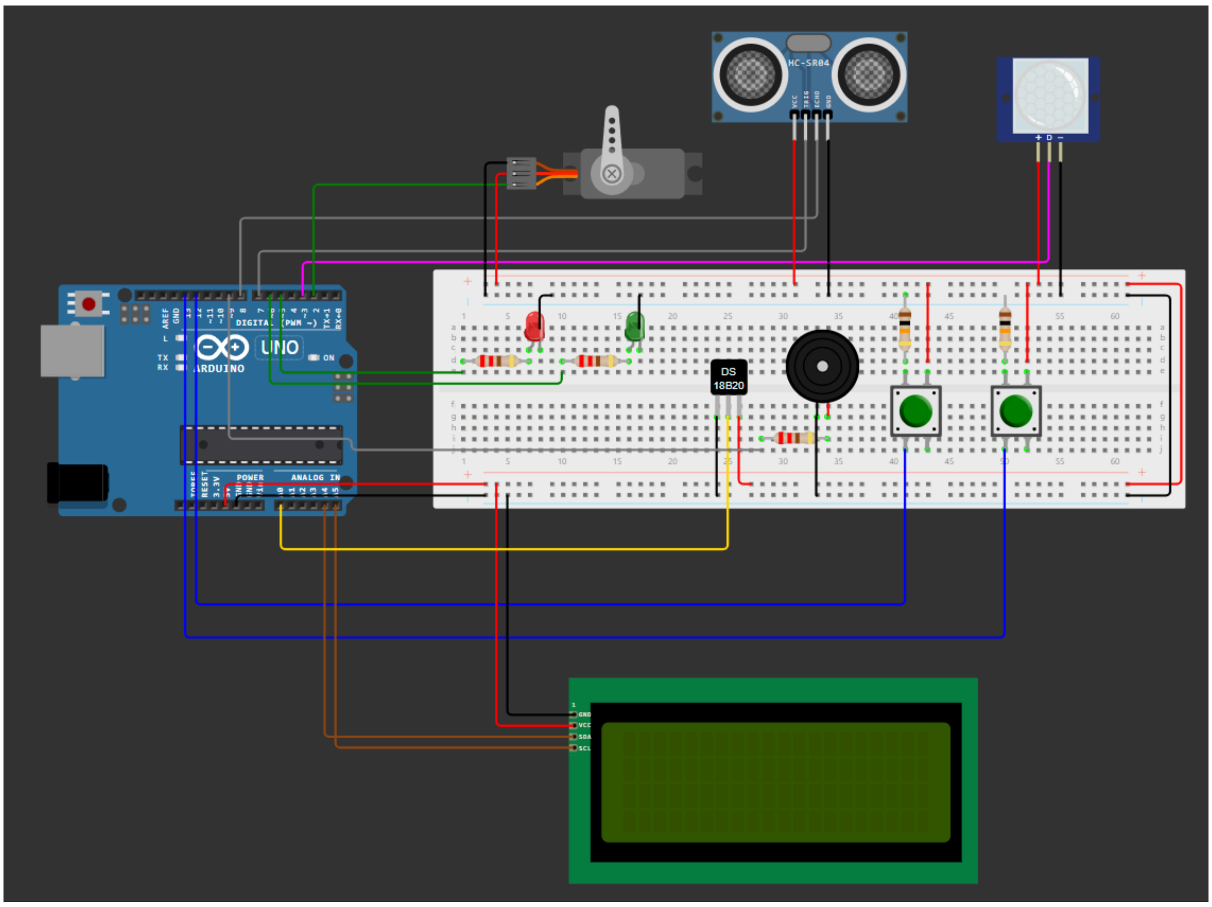
Assignment 02 - *Smart Waste Disposal System*

Andrea Baldazzi 0001071149

Arduino subsystem

Breadboard view



Architecture

The system is realized with synchronous Finite State Machines, which means that there is a scheduler that periodically executes each Task. The scheduler period is set to be the greatest common divisor of all the Tasks periods.

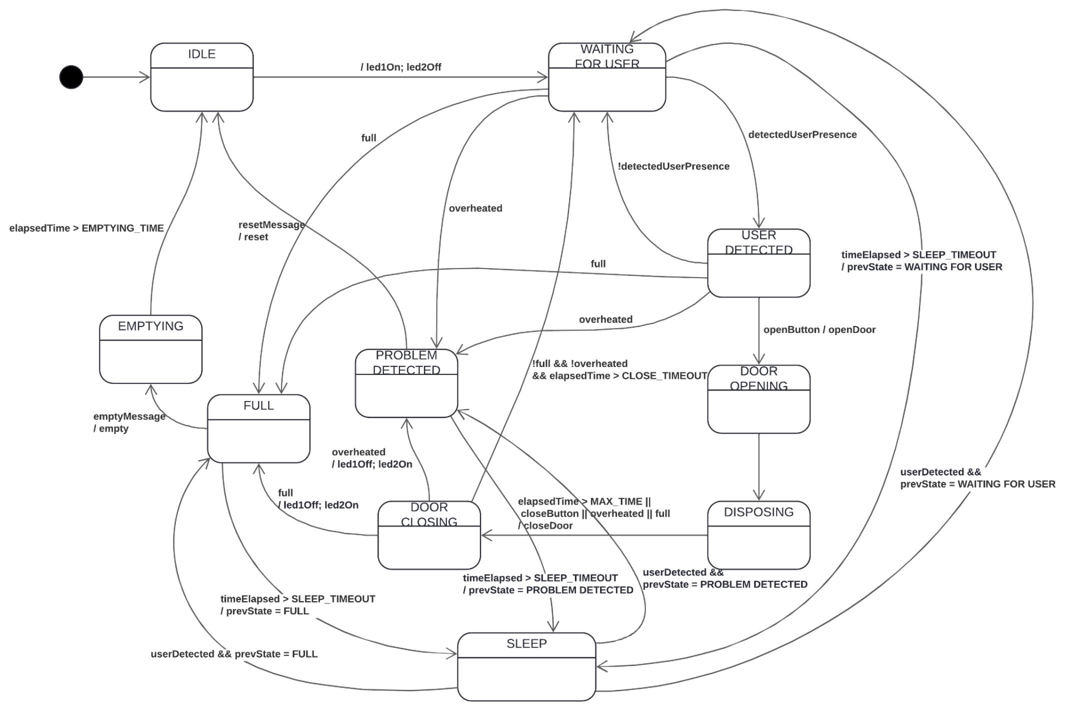
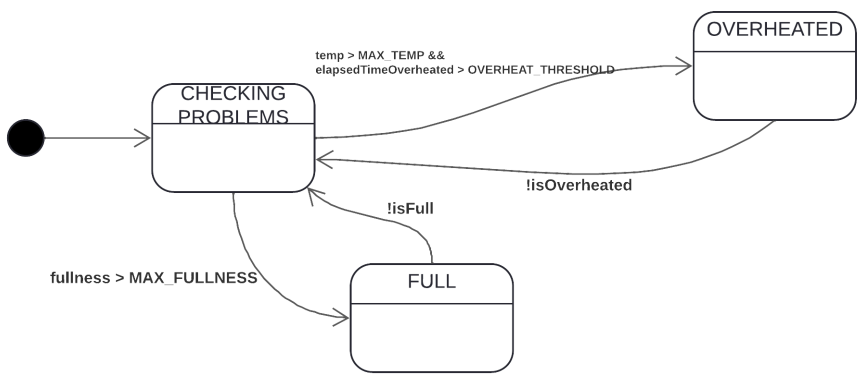
Tasks

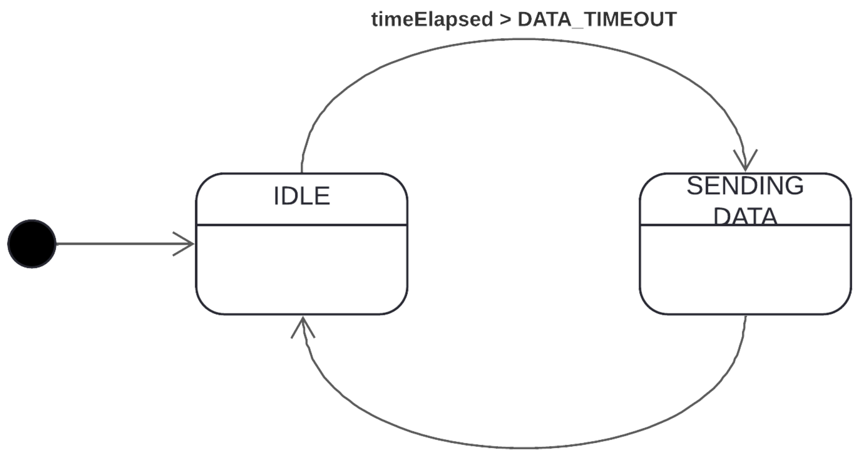
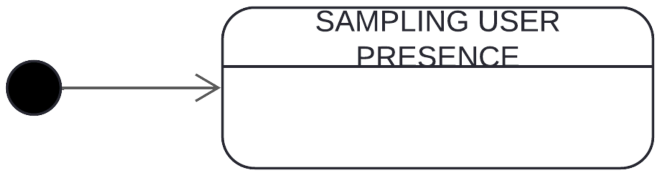
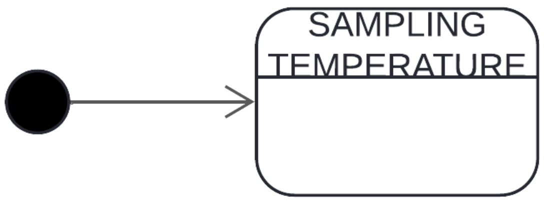
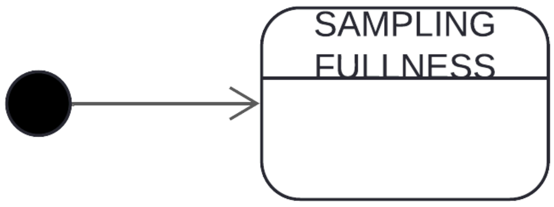
My solution solution is organised into 7 tasks:

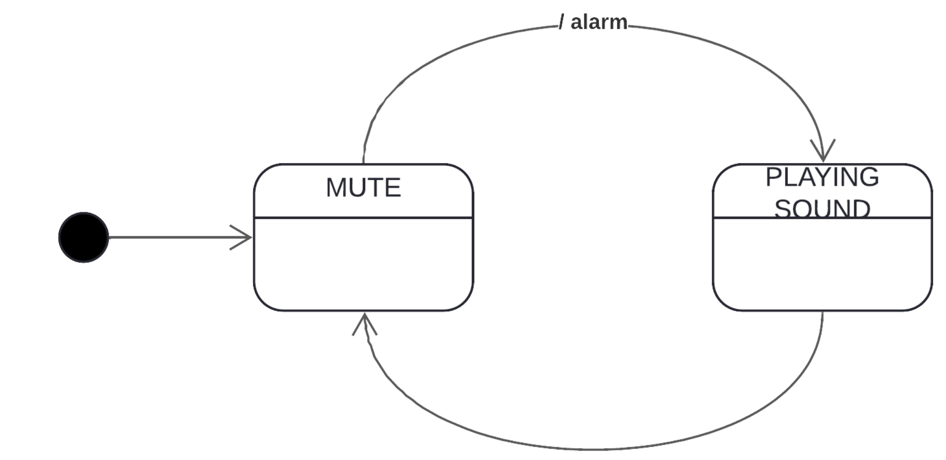
- **WorkflowTask**
  - main task, managing the global workflow of the waste disposal system
  - responsible for the sleeping process
  - consults the WasteDisposalSystem to see if CheckTask has raised an alert, then proceeds solving it
- **CheckTask**
  - task periodically checking for error in the system
  - consults the WasteDisposalSystem to check if the container is full or overheated, making him aware of it
- **TelemetryTask**
  - task responsible for communicating with the pc system
  - periodically sends system info (status, fill percentage, temperature)
- **UserPresenceTask**
  - task telling the system to sync for user presence
- **TemperatureTask**
  - task telling the system to sync the temperature reading
- **FullnessTask**
  - task telling the system to sync the container waste level
- **WarningTask**
  - task beeping an alarm sound when activated
  - activated and deactivated by WorkflowTask

The idea of having these three sampling tasks (TemperatureTask, FullnessTask and UserPresenceTask) is to make the system update the sampling from the sensors easily programmable. That can be achieved by changing the task period. By doing this way, the other tasks don't need to read values directly from the hardware multiple times per tick, instead they get the info they want by consulting the system object.

FSMs representing the behaviour for each task:

WorkflowTask *period: 100 ms*	 IDLE / led1On; led2Off WAITING FOR USER full detectedUserPresence !detectedUserPresence overheated elapsedTime > EMPTYING_TIME resetMessage / reset USER DETECTED timeElapsed > SLEEP_TIMEOUT / prevState = WAITING FOR USER full EMPTYING overheated openButton / openDoor PROBLEM DETECTED !full && !overheated && elapsedTime > CLOSE_TIMEOUT DOOR OPENING FULL emptyMessage / empty overheated / led1Off; led2On userDetected && prevState = WAITING FOR USER full / led1Off; led2On DOOR CLOSING elapsedTime > MAX_TIME || closeButton || overheated || full / closeDoor DISPOSING timeElapsed > SLEEP_TIMEOUT / prevState = PROBLEM DETECTED userDetected && prevState = PROBLEM DETECTED timeElapsed > SLEEP_TIMEOUT / prevState = FULL SLEEP userDetected && prevState = FULL
CheckTask *period: 100 ms*	 OVERHEATED temp > MAX_TEMP && elapsedTimeOverheated > OVERHEAT_THRESHOLD CHECKING PROBLEMS !isOverheated !isFull fullness > MAX_FULLNESS FULL

TelemetryTask *period: 100 ms*	
UserPresenceTask *period: 100 ms*	
TemperatureTask *period: 100 ms*	
FullnessTask *period 200 ms*	

WarningTask *period 500 ms*	

Other components

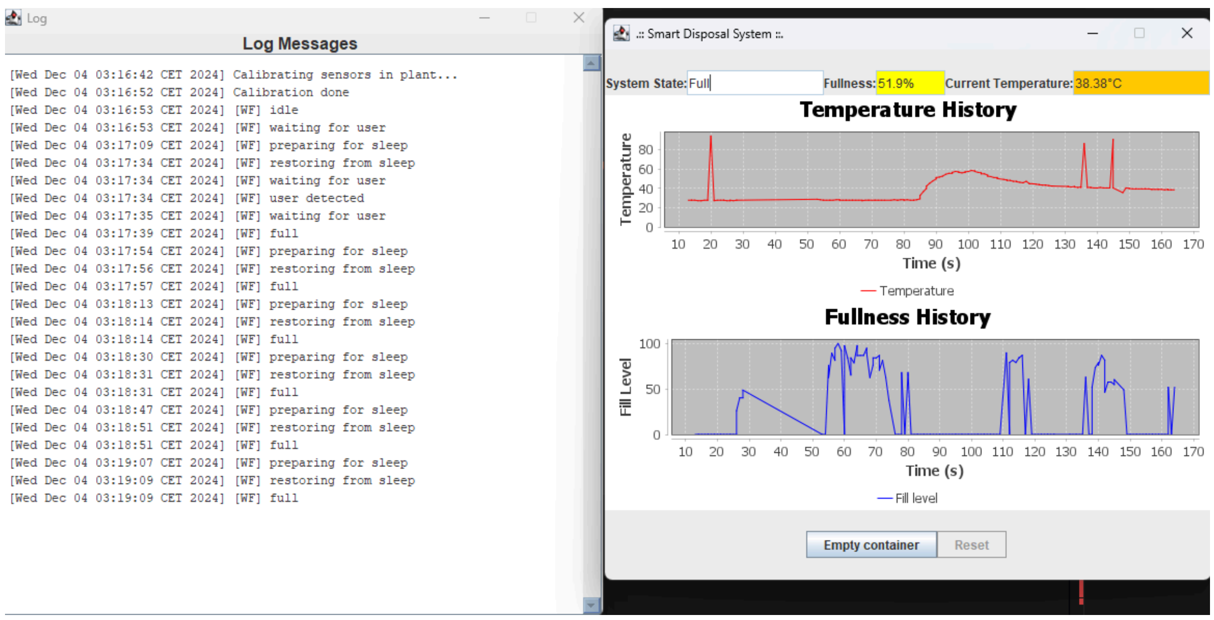
Besides tasks, there are other important components of the software:

- **WasteDisposalSystem** (*src/model/WasteDisposalSystem.h, src/model/WasteDisposalSystem.cpp*)
  - main entity of the model, representing the system, shared among tasks
  - it works with internal states used to tell tasks about the current situation of the system
  - it's the only one who contains the objects associated with hardware components (except buttons and lcd)
  - when required to sync input readings from one of the three sampling tasks, it operates physical sensors and stores values locally for other tasks to use
- **UserConsole** (*src/UserConsole.h, src/UserConsole.cpp*)
  - component responsible for input and output interactions with the user
  - operates physical LCD display and buttons

PC dashboard subsystem

The pc dashboard is realized in Java, using the JSSC library to communicate serially with arduino through the UART protocol
Main components of the dashboard subsystem:

- **MonitoringAgent**
  - active component responsible for waiting for messages from the serial line with Arduino
  - redirects info to the log or the dashboard depending on appropriate prefixes
- **DashboardView**
  - main view of the pc program, from which the system admin can control the container
  - displays system current state, fullness and temperature
  - has empty and reset buttons the allow the admin to fix the system
  - plots fullness and temperature values in two graphs creating an history
  - messages with prefix “wd:” are routed here from the MonitoringAgent
- **LogView**
  - panel for displaying logging info coming from the system
  - messages with prefix “lo:” are routed here from the MonitoringAgent
- **SmartWasteDisposalSystemDashboardController**
  - manage events from the view, sending messages to Arduino



```
[Wed Dec 04 03:16:42 CET 2024] Calibrating sensors in plant...
[Wed Dec 04 03:16:52 CET 2024] Calibration done
[Wed Dec 04 03:16:53 CET 2024] [WF] idle
[Wed Dec 04 03:16:53 CET 2024] [WF] waiting for user
[Wed Dec 04 03:17:09 CET 2024] [WF] preparing for sleep
[Wed Dec 04 03:17:34 CET 2024] [WF] restoring from sleep
[Wed Dec 04 03:17:34 CET 2024] [WF] waiting for user
[Wed Dec 04 03:17:34 CET 2024] [WF] user detected
[Wed Dec 04 03:17:35 CET 2024] [WF] waiting for user
[Wed Dec 04 03:17:39 CET 2024] [WF] full
[Wed Dec 04 03:17:54 CET 2024] [WF] preparing for sleep
[Wed Dec 04 03:17:56 CET 2024] [WF] restoring from sleep
[Wed Dec 04 03:17:57 CET 2024] [WF] full
[Wed Dec 04 03:18:13 CET 2024] [WF] preparing for sleep
[Wed Dec 04 03:18:14 CET 2024] [WF] restoring from sleep
[Wed Dec 04 03:18:14 CET 2024] [WF] full
[Wed Dec 04 03:18:30 CET 2024] [WF] preparing for sleep
[Wed Dec 04 03:18:31 CET 2024] [WF] restoring from sleep
[Wed Dec 04 03:18:31 CET 2024] [WF] full
[Wed Dec 04 03:18:47 CET 2024] [WF] preparing for sleep
[Wed Dec 04 03:18:51 CET 2024] [WF] restoring from sleep
[Wed Dec 04 03:18:51 CET 2024] [WF] full
[Wed Dec 04 03:19:07 CET 2024] [WF] preparing for sleep
[Wed Dec 04 03:19:09 CET 2024] [WF] restoring from sleep
[Wed Dec 04 03:19:09 CET 2024] [WF] full
```

System State: Full
Fullness: 51.9%
Current Temperature: 38.38°C