

Smart Car Wash System Report

Javid Ameri, Matilde D'Antino, Federico Diotallevi

1 Introduction

The smart car wash system simulates an automated car washing system. The system comprises various hardware components such as presence and distance sensors, a servo motor, an LCD display, a start button, LEDs, and an temperature sensor. The system communicates with a PC through a serial connection for monitoring and maintenance interventions.

2 System Architecture

The system is divided into two main areas: Check-In/Out and Washing. It utilizes tasks and synchronous Finite State Machines (FSM) to manage the phases of the car washing process. State management is implemented through a bridge class containing a state variable accessible externally through the `getState()` method.

2.1 System States

The identified states are:

- **CAR_WAITING:** The initial state when the car is waiting.
- **WELCOME:** The state indicating the welcome phase, triggered when a car is detected by the motion sensor.
- **GATE_OPENING:** The state when the car wash gate is opening and closes when the car is at the correct distance.
- **READY_TO_WASH:** The state when the system is ready to start washing, waiting for the user to push the start button.
- **CAR_WASHING:** The state representing the active car washing process.
- **WASHING_COMPLETED:** The state indicating that the car washing is completed.
- **MAINTENANCE:** Maintenance state.

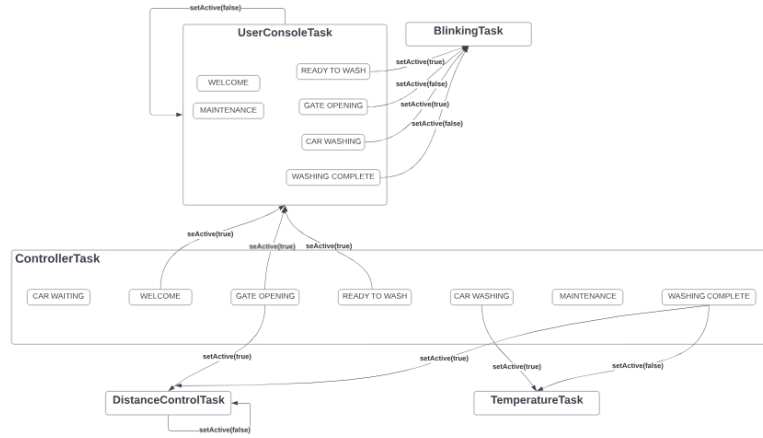


Figure 1: Task Activation Dependency Diagram

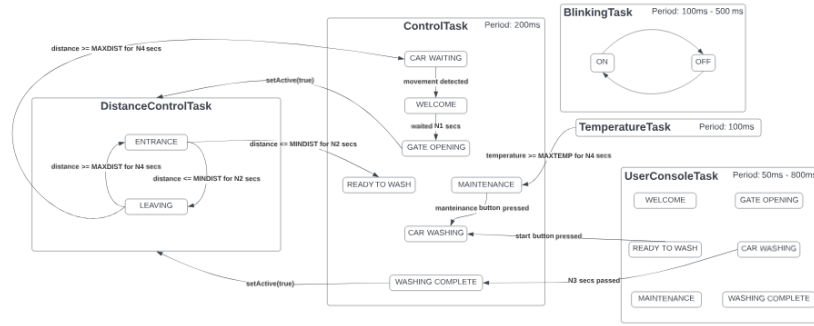


Figure 2: Finite State Machine Diagram

3 Code Implementation

The Arduino code is implemented using task-based architectures and synchronous FSMs. Serial communication allows system monitoring from the PC Console Dashboard. State management is handled by the bridge class, which exposes the `getState()` method for state consultation by all classes.

4 Implemented Tasks

Five tasks have been developed:

4.1 Blinking Task

This task is initialized with an LED, which is the subject of blinking. The task contains two internal states, `ON` and `OFF`, which alternate to turn the LED on and off. It is a periodic task with an adjustable period through a public method.

4.2 DistanceControl Task

This task contains two internal states, `ENTERING` and `LEAVING`. It implements gate control using a servo motor and a proximity sensor. The gate opens in both states and closes if:

1. The state is `ENTERING`, and the car is close enough to the sensor for $N2$ seconds.
2. The state is `LEAVING`, and the car remains far enough from the sensor for $N4$ seconds.

When the gate is closed, if the state is `ENTERING`, it is changed to `LEAVING`, and the Bridge state is changed to `READY_TO_WASH`. If the state is `LEAVING`, it is changed to `ENTERING`, and the Bridge state is changed to `CAR_WAITING`.

4.3 Temperature Task

The `TemperatureTask` is a periodic task activated during the `CAR_WASHING` state in the intelligent car washing system. This task periodically queries the DHT temperature and humidity sensor to obtain the current temperature within the washing area. During the car washing process, it monitors the temperature to ensure suitable washing conditions. The obtained temperature data is transmitted to the dashboard for monitoring. If the measured temperature exceeds a predefined threshold (`MAXTEMP`), the `TemperatureTask` suspends the washing process and communicates to the dashboard that maintenance is required.

4.4 UserConsole Task

The `UserConsoleTask` is a task responsible for managing the user interface and providing feedback to the user during different states of the car washing process. It uses an LCD display, LEDs, and a blinking task for visual indications. The task is designed to display relevant information based on the current state of the system, such as welcoming messages, instructions, and progress during the car washing process. It interacts with the bridge to retrieve the current state and adjust its behavior accordingly.

4.5 Controller Task

The `ControllerTask` is a crucial component in coordinating the operation of the smart car wash system. It interacts with various tasks, including the `UserConsoleTask`, `TemperatureTask`, and `DistanceControlTask`. This task plays a central role in orchestrating the entire car washing process.

Initialization

The `ControllerTask` is initialized with references to the `UserConsoleTask`, `TemperatureTask`, `DistanceControlTask`, and the bridge. During initialization, it sets the initial state to `CAR_WAITING`.

State Handling

The `tick()` method of the `ControllerTask` manages system states and triggers corresponding actions. It activates and deactivates other tasks based on the current state read from the bridge.

5 Schema

Below is the schema of the Arduino-based smart car wash system.

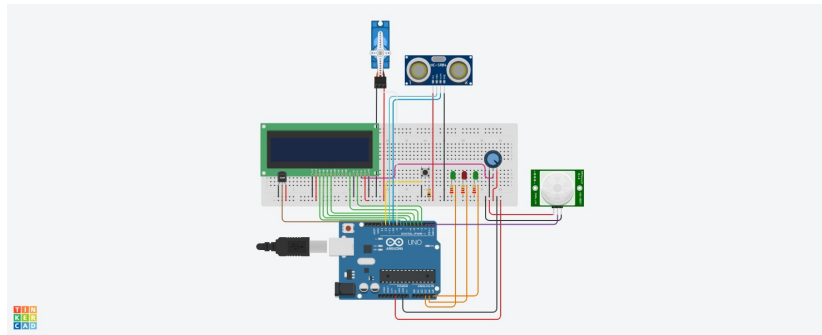


Figure 3: Schema