Smart Car Washing System IoT

Leonardo Tassinari,

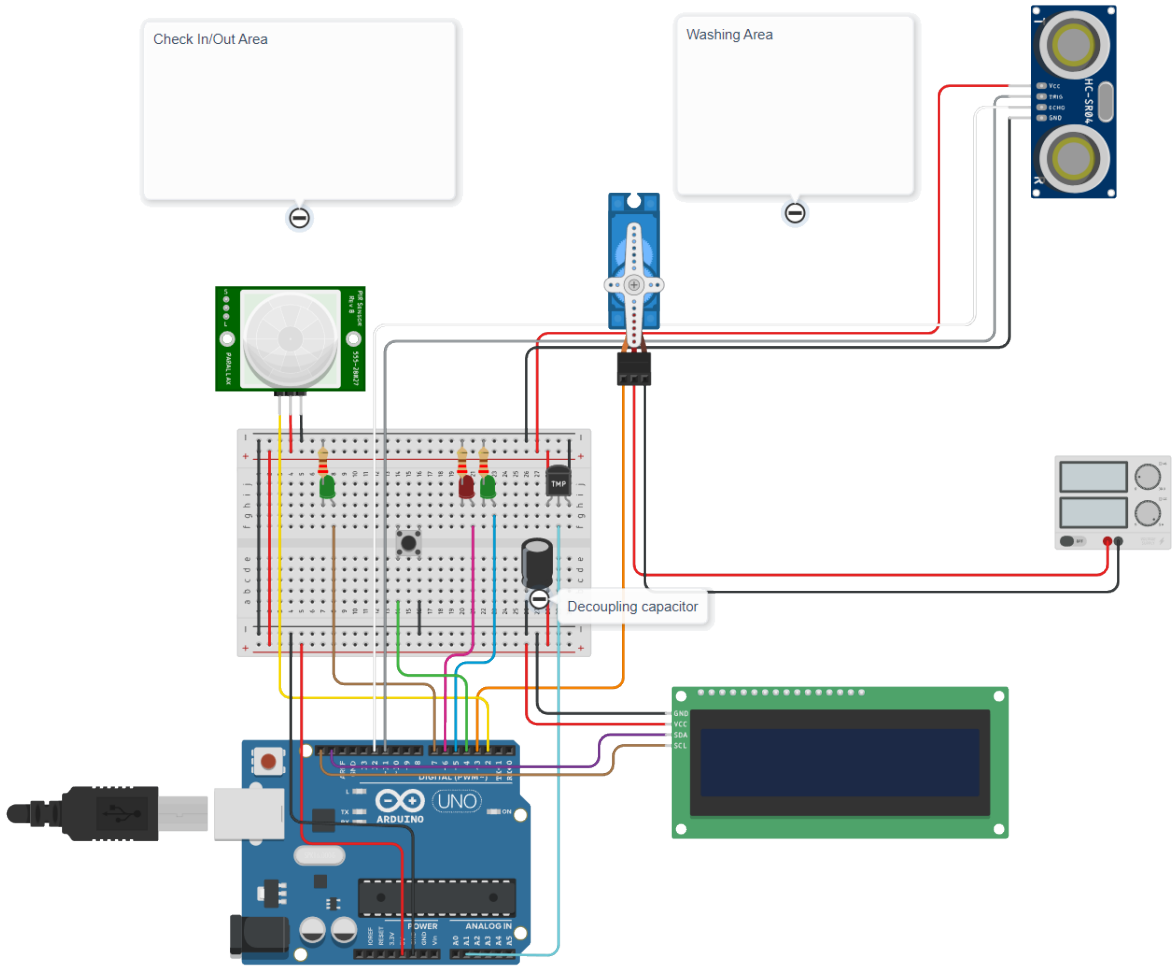
Michele Olivieri

December 3, 2023

Contents

1 Circuit

The circuit schematic is as follows:



Device	pinout
Temperature sensor	A1
Pir	2
L1	7
L2	6
L3	5
Servo	3
Sonar trig	11
Ssonar echo	12

Since the servo motor might draw more current than the 150mA offered by arduino pins it's advisable to feed the servo with external power. All the elements attached to the power line create noise that hinder the analog temperature sensor's readings.
To dampen this effect it is possible to add decoupling capacitors (10-100 uF) to filter a bit those variation.

2 Software Architecture

2.1 Arduino

The program running on arduino is written in C++ using the wiring framework and the platformIo utility.
It is managed by a syncronous scheduler, it is triggered by a timer and launches non cooperative tasks.
Each task period has been chosen accordingly to its operative requirements, and is a multiple of the base period of the scheduler. Some empiric tests have been made to ensure that no overrun happens in any scenario.
The work tree of the program is divided in:

- Sensors, containing interfaces (abstract classes) and implementations for the devices.
- actuators, containing interfaces (abstract classes) and implementations for the devices.
- System, containing the scheduler and the task interface.
- Task, containing all the tasks.

On the root are present:

- The main file, entry poit of the program, where the scheduler and the tasks are instantiated.
- CarWash class, representing the domain of the system with all the shared variables.
- Config, configuration file to tune all the parameters of the program.

2.2 Rust

The rust application uses "egui", a simple, fast, and highly portable immediate mode GUI library. The other main crate used is "Serialport" for serial communication, a new thread is created to listen to the selected port, it expects a **Json** with the parameters of the CarWash class, it then parses it and records the data inside the program. To do so also mutexes are used to ensure no race conditions since also the gui thread is using the same data.
It has been tested both on Windows and MacOs.

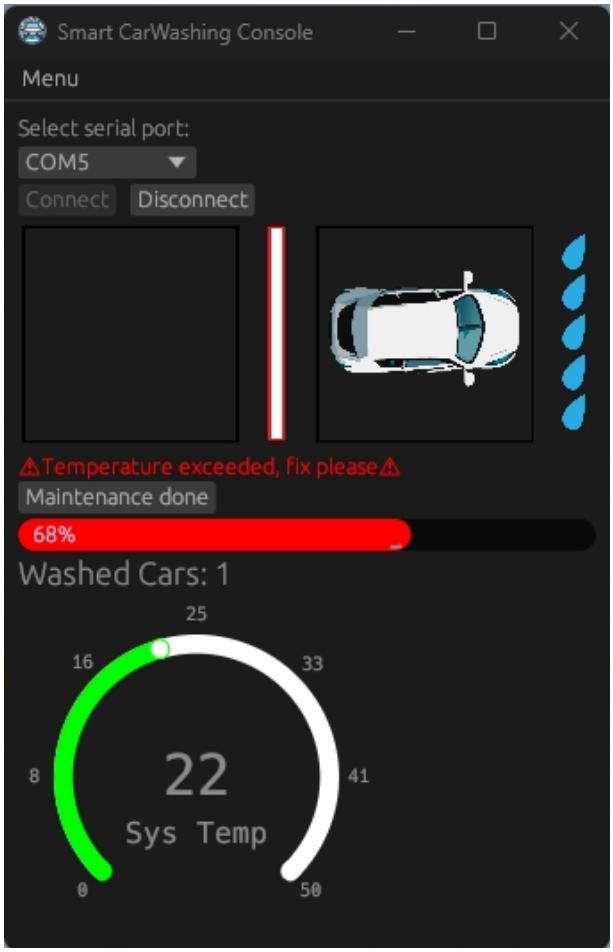
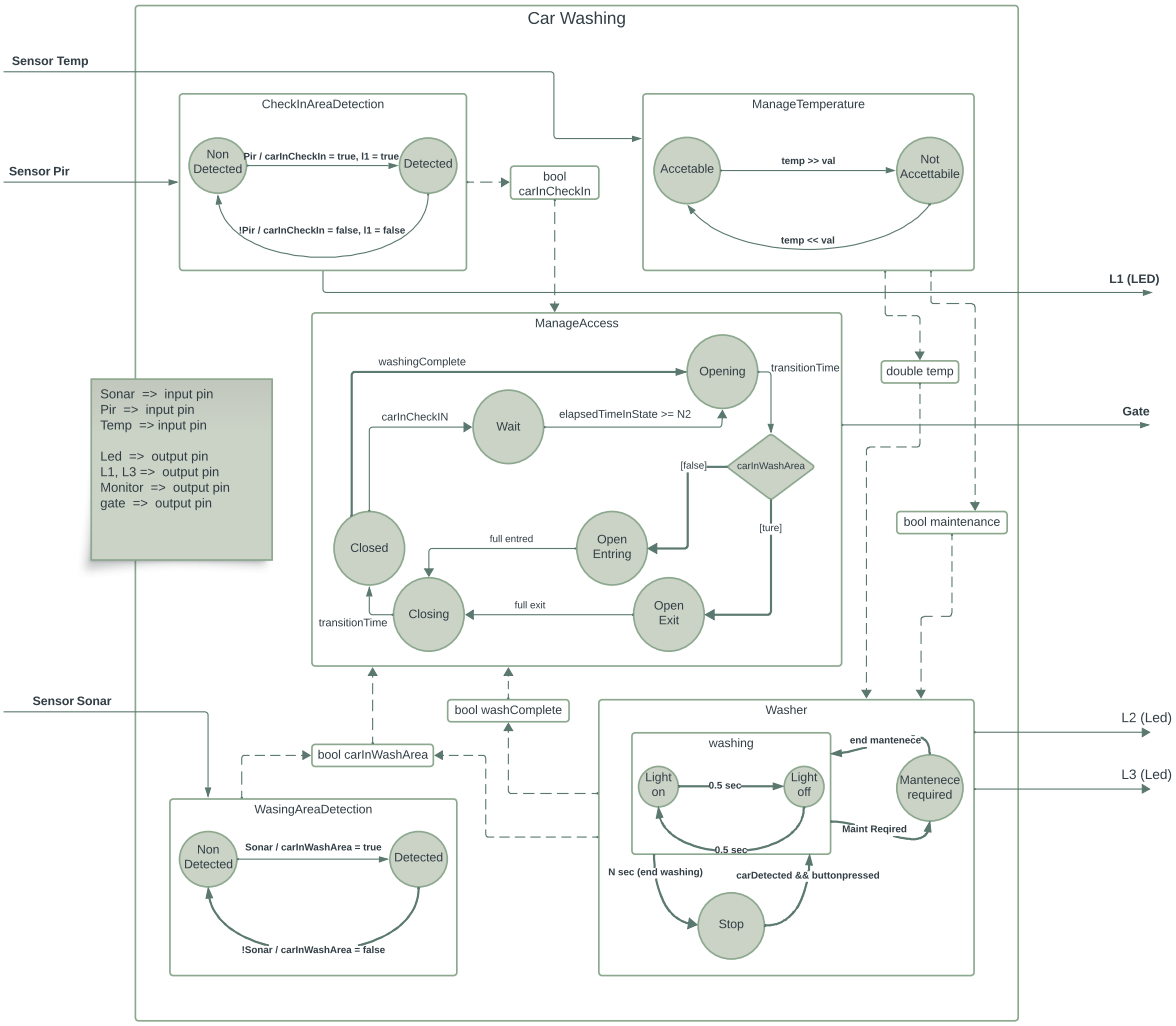


Figure 1: Screenshot of the dashboard ui

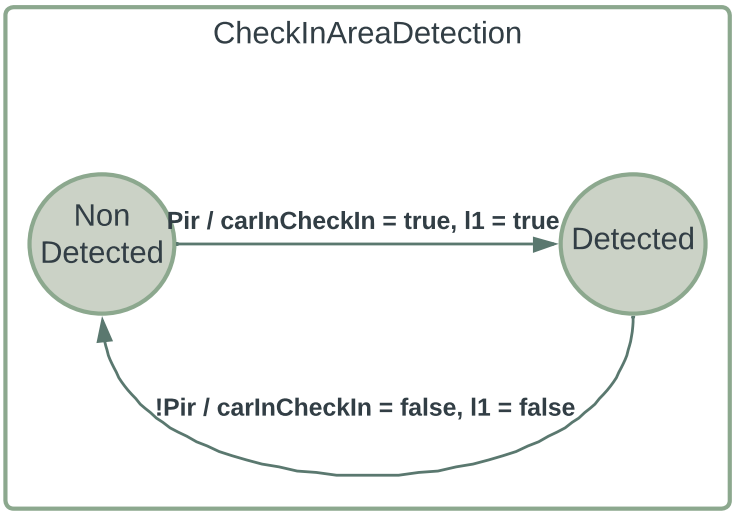
3 General diagram



4 Tasks

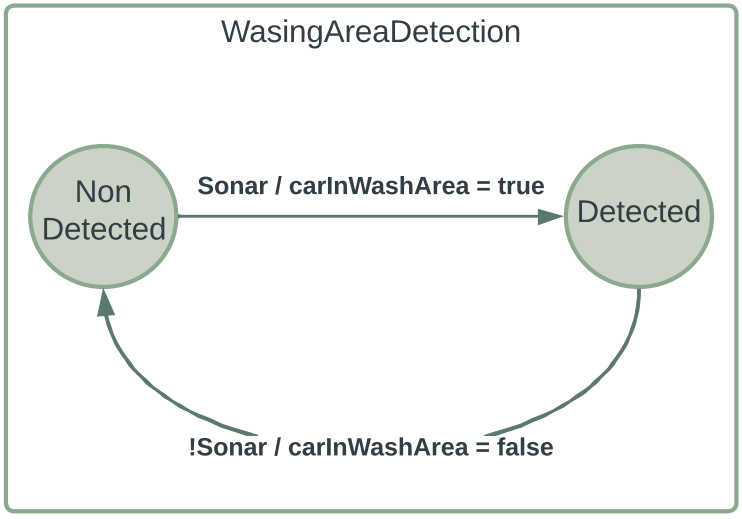
4.1 checkInDetection

This task manages the detection of a car inside the check in area with a pir sensor and the state of the L1 led.



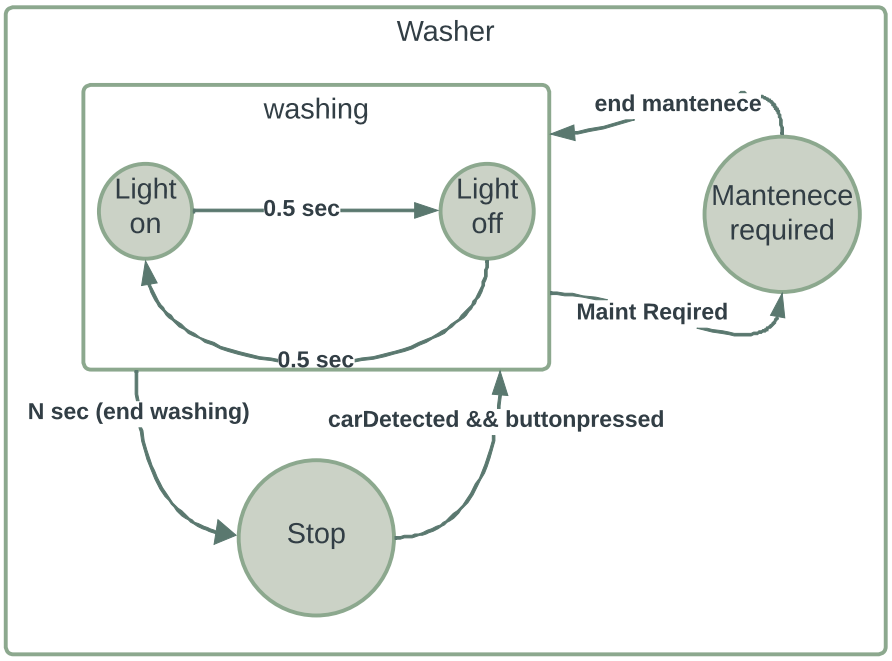
4.2 washingAreaDetection

This task manages the detection of a car inside the washing area with a sonar sensor.



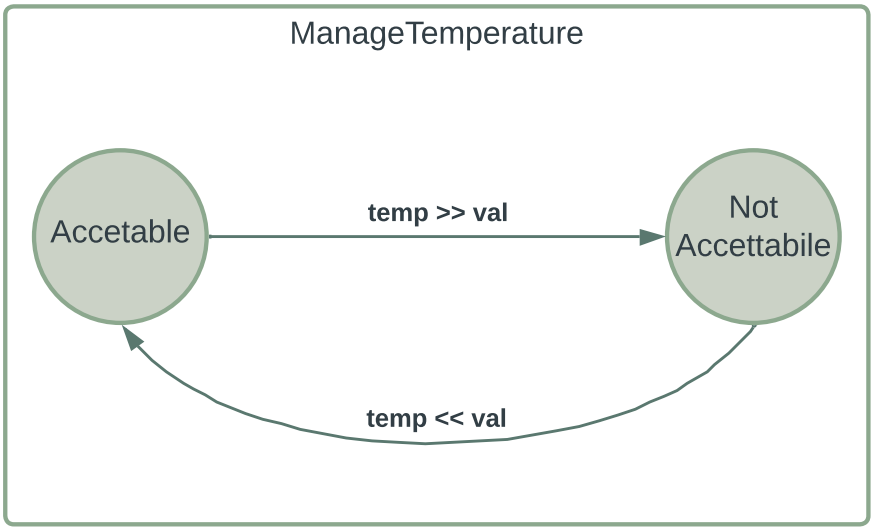
4.3 washer

This task manages the washing process, L2 and L3 leds.



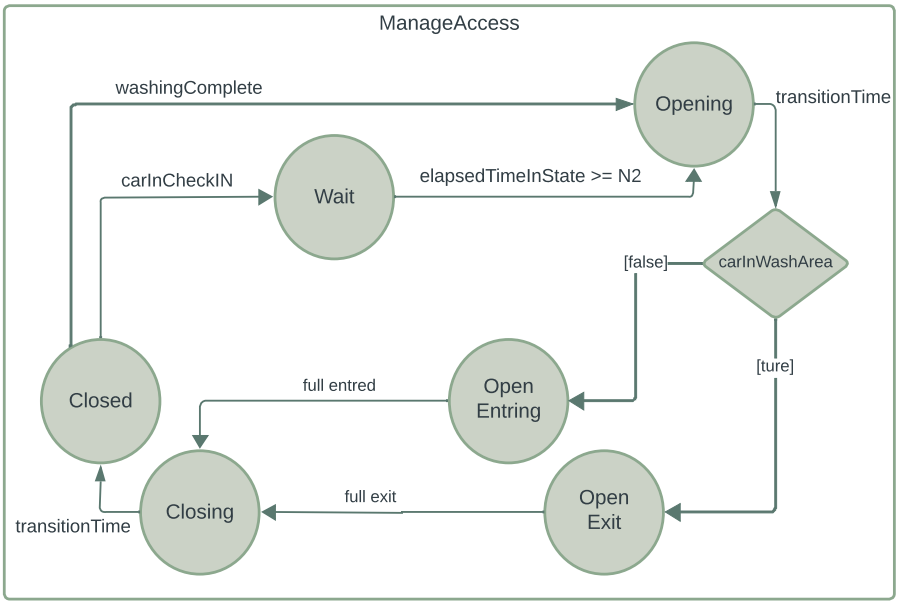
4.4 manageTemperature

This task checks the temperature with an analog sensor and records it, it also signals if it is over the limit.



4.5 manageAccess

This task manages the position of the bar.



4.6 displayFeedback

This task displaies the appropriate message on the display.

4.7 communicator

This task is responsible of communicating via serial with the dashboard.

4.8 sleepTask

This task triggers the sleep when the situation requires it.