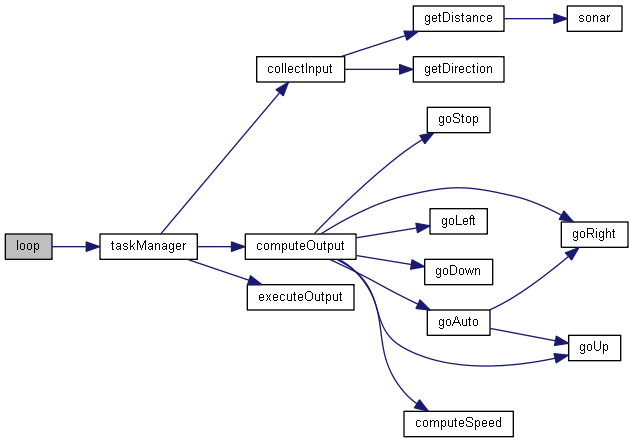
***FUNCTIONALITY « WiFi controlled Car »***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| car.docx | | | | **History** | |
| **Issue status**  (Index) | **Maturity/Date**  (draft/invalid/valid)  (dd-mmm-yyyy) | **Author**  Department | **Check/Release**  Department | | **Description** |
| 1.0 | Valid  13.09.2018 | A. Tosa  Intern |  | | Created to describe the functionality of the car. |

****

**Table of contents:**

1. **Purpose**
2. **Realization constraints and targets**
3. **Architecture**
   1. External interaction
   2. Functionality’s internal breakdown
   3. Circuit sketch
   4. Modules description
   5. Possible improvements
4. **Bibliography**

**1 Purpose**This document represents the design of the car’s software and hardware component.

# 2 Realization constraints and targets

The car should respond to the commands and be able to detect the obstacles in front of it, in order to not hit them in either control mode (automatic or not).

**3 Architecture**

**3.1 External interaction**The interaction with the car is done through a web interface which can be accessed either by connecting to the ESP8266 access point or through the wifi network to which the ESP connects. The web interface consists of five buttons with the following functionalities: the middle button turns on auto-pilot mode, which is turned off by pressing any button, and the adjacent buttons move the car in a direction, depending on the position of the button.

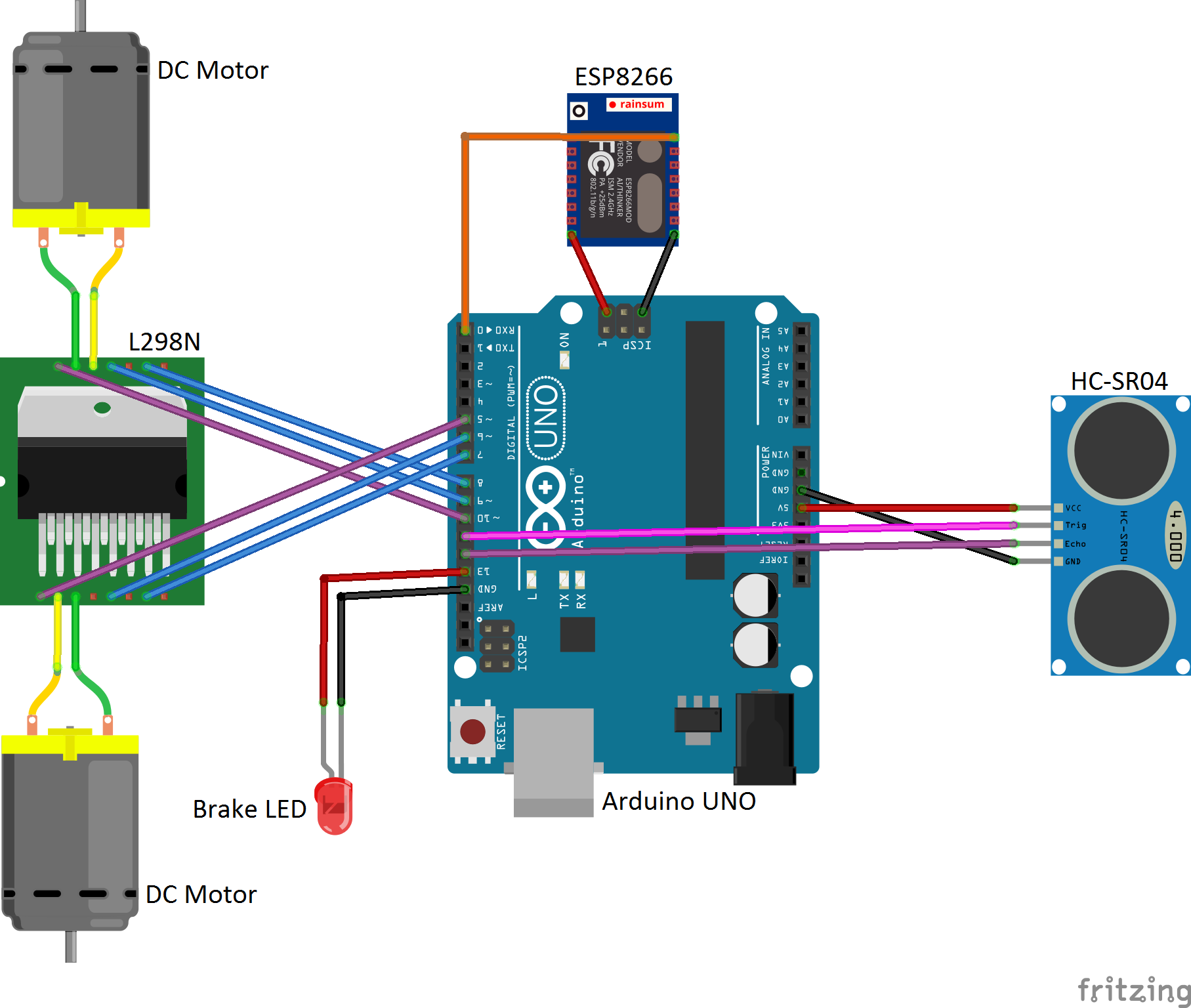
**3.2** **Functionality’s internal breakdown**

The car consists of a chassis on which are mounted two **DC**, an **Arduino UNO**, an **ESP8266**, an Ultrasonic **HC-SR04** sensor and a **L298N** Dual Full Bridge Driver.

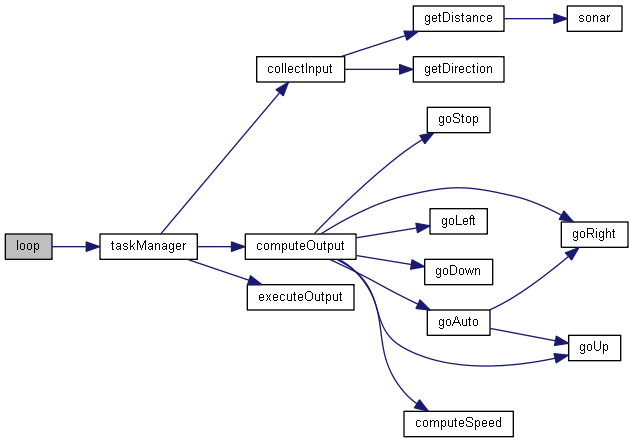
The ESP acts both as an access point and a station to which the user can connect. It displays the control interface and handles the user requests, which it the sends through serial communication to the Arduino.

The Arduino receives the commands from the ESP, and engages the motors accordingly. It checks using the ultrasonic sensor that there are no obstacles. If there are obstacles it tries to avoid them by turning to the right if in auto mode, otherwise it stops going forward.

The L298N motor driver receives power from an external source and controls the motors depending on the inputs it receives from the Arduino.

**3.3 Circuit sketch**

**3.4 Modules description**



* bool getDistance(bool currDistance)

Determines if the distance to the object in the front of the car is not less than the minimal distance. It takes multiple measurements in order to be sure they are accurate.

* int getDirection(int currDirection)

Receives the code for a direction through serial communication.

* int computeSpeed(int currSpeed, int typeOfMovement)

Determines the speed of the car depending on the current speed and the type of movement. If moving straight, it accelerates. If on auto, the car moves slower.

* void goAuto()

Controls the car in auto mode. The car moves forward until there is an obstacle. Then it starts rotating right until an opening big enough is detected and the car starts moving toward it. If the car is close to on object, it will rotate further in order to not hit the object with a wheel.

* void startWiFi()

Starts the ESP in AP and STA mode. It starts the access point with the name “cascaval” and the tries to connect to the chosen wifi network.

* void webSocketEvent(uint8\_t num, WStype\_t type, uint8\_t\* payload, size\_t length)

Receives information from the web socket and handles it accordingly. If the information is of type text, it will be sent to the Arduino.

* void startServer()

Starts the http server and defines which function shall be called depending on which location is accessed.

**3.4 Possible Improvements**

A first improvement would be adding more distance sensors for a better detection of the obstacles. They could be mounted above each wheel, at an angle to cover a bigger area. Another solution would be to mount the sensor on a servo/stepper motor to spin it around and cover a bigger area.

Another important improvement would be adding a battery on the car, so it wouldn’t need to be plugged in to an external power source.

The motors could also be improved since they are pretty weak and have some wear into them.

An improvement regarding the software would be displaying some information gathered by the car on the web interface.

**4 Bibliography**

* <https://circuits4you.com/2018/02/05/esp8266-arduino-wifi-web-server-led-on-off-control/>
* <http://www.martyncurrey.com/arduino-to-esp8266-serial-commincation/>
* [https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-L298Nn-pwm-h-bridge/](https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/)
* <https://tttapa.github.io/ESP8266/Chap14%20-%20WebSocket.html>
* <https://blog.miguelgrinberg.com/post/building-an-arduino-robot-part-i-hardware-components>