

# STREAMING ROBOT

Todoran Denisa, Luca Daniel, Muntean Dorian, Politehnica University of Timisoara May, 2018

## 1 Repository

*The GitHub used for the storage of documents history, schematics and any other information is:*

Schematics, diagrams and codebase are contained under the following git repository:

<https://github.com/tdeny96/MS>

## 2 User requirements

1. The robot should provide clear images of its surroundings, that should be visible from the user's phone or computer screen;
2. The system should be open for extension; to save the data that's been recorded in a certain repository from where they can be accessed at any time;
3. The robot should start and be manipulated via a Web interface;
4. The robot and the streaming process must be done in a zone with access to the internet;
5. The robot must not be used on ground that has a lot of cracks to avoid the risk of damaging it;

## 3 System overview

The overview of the system is as depicted in the figure below: 1.

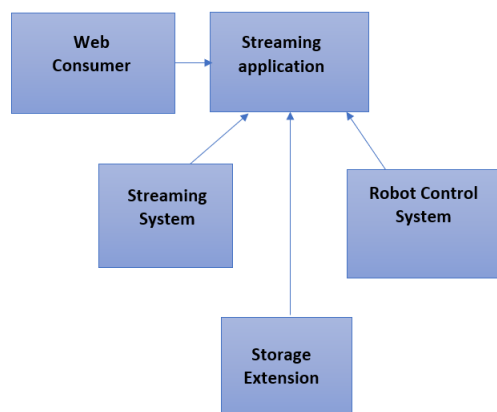


Figure 1: System overview diagram

The Streaming application has the role of starting the system and allow the user to manipulate it, by making the robot move left, right, back etc.

The Streaming System has the purpose of filming the surrounding and sending them to either the mobile or Web user.

The Robot Control System will control the movement of the robot according to the commands given by the users.

Web consumer provides a UI interpretation for the data transmitted by the streaming system. This view is accessed within a Web browser.

## 4 Circuit design

The hardware view of the system is depicted below: 2.

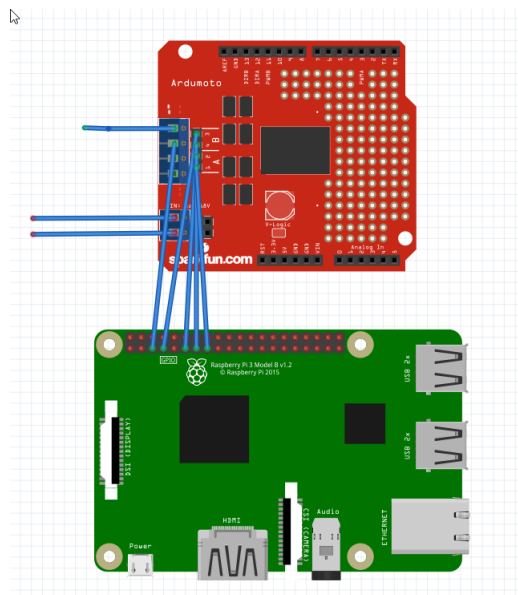


Figure 2: Circuit schematic

Raspberry Pi 3 provides support for quick prototyping. We will use the one-wire interface it has and the possibility to communicate with other devices over the Internet.

As the figure could not be made so it represents the real model, it cannot be seen that there are two motors both connected the L298N driver through wires and the board is connected to the 12V.

## 5 Software design

The software components and data flow directions are depicted below: 3.

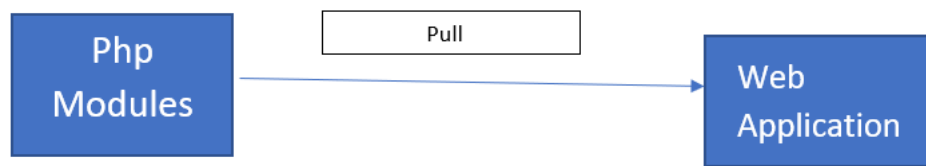


Figure 3: Software entities involved

### 5.1 PHP modules

Camera\_rotate.php: commands the movement of the robot depending on the user input.

Camera.php: the file where we can see the streaming done by the camera.

Camera.js: file that continuously listens to the input of the user.

### 5.2 Web Application

In the web application, the user can both access the streamed data and control the movement of the robot by holding down the desired keys that represent the direction he wants the robot to head off to.

## 6 Results:

So far, we managed to:

- have the camera stream any time as long as there is internet connection;
- connect the raspberry pi to the internet and create a webpage for it;
- access the webpage on any device that is connected to the same WIFI as the raspberry pi;

We plan to:

- have the robot move with commands given over the internet;
- have the device store the streamed videos on a GitHub account;

## 7 References

1. Fritzing [last seen: May 2018], <http://fritzing.org/>
2. <https://www.hackster.io/whitebank/raspberry-pi-remote-control-car-camera-a70>
3. <http://www.instructables.com/id/L298N-Dual-H-Bridge-Raspberry-Pi/>
4. <https://circuitdigest.com/microcontroller-projects/raspberry-pi-remote-control-car>
5. <http://webiopi.trouch.com/INSTALL.html>
6. <http://www.instructables.com/id/Raspberry-Pi-2-WiFi-RC-Car/>