

# Browser Based Remote Control of Raspberry Pi enabled RC Car



## Contact:

Jonas Karlsson [jonas.karlsson@kau.se](mailto:jonas.karlsson@kau.se)  
Department of Computer Science,  
Karlstad University

## Background:

We have a traxxas RC car with a raspberry pi, GoPro Hero cam and a 5G Modem. To communicate with the car over the network for video streaming, issuing control commands, etc., we have developed a Web Real-Time Communication (WebRTC) framework. The framework is built on Pion (<https://pion.ly/>). A foundation library for controlling the servo motors (<https://gobot.io>) are also available.

The car should be controlled by an operator using a web-browser. The operator should be able to modify the throttle (forward and reverse), steer left and right and press the brakes. A remote-controlled car must also contain a security system that in a safe manner stops the car if contact with the operator is lost.

## Task:

Integrate the raspberry pi GPIO PWM pins with the Electronic Speed Control (ESC, Traxxas xl-5 [MODEL 3018R](#)) of the [RC CAR](#) and the steering servo using the gobot library. This task can be divided in a physical part that require electrical/mechanical knowledge of servomotors and a software development part that require knowledge about golang.

The physical setup should be easily removable (=no soldering and no unscrewing of cables) to restore the original function of the car (or be able to run them in parallel).

Furthermore, a control scheme and a “safety system” should be designed and developed. The control scheme should allow the operator via the keyboard to run the car forward and reverse with different speeds and different angle on the steering as well as push the breaks.

The safety systems should do an emergency break if contact with the operator is for some reason is lost.

Bonus tasks:

- Control should be done via a gamepad

- Safety system should slow down if a “large” object is within a specific distance (detected with a distance sensor)

## **Epic:**

The system will build upon previous project(s): <https://git.cs.kau.se/research/dwr/picar1> that have done the basic work.

The exact epics and sub-task will be discussed and evaluated with the development team but should consist of the following:

- Physical connections and Controlling of the servo and ESC
- Control Scheme design and implementation
- Safety scheme

## **Legal and Technical Requirements:**

The source code produced in this project should be published under a copyleft license (ie GPL 3).

Focus will be on product readiness and stability. All code must adhere to common best practices and follow standardized language conventions.

Code checks, tests and build procedures should be automated (gitlab-ci) as much as possible and the outcome (build package) should directly be installable in the host OS (Raspberry Pi debian bulleseye).

The CS department at KaU will provide the car, raspberry pies, a public facing server for the webtrc parts and provide the repositories and help with setting up repositories.