
Rover Documentation

Browser-based Camera Stream & Drive Control

Version

July 30, 2017

Implementation

Mustafa Özcelikörs

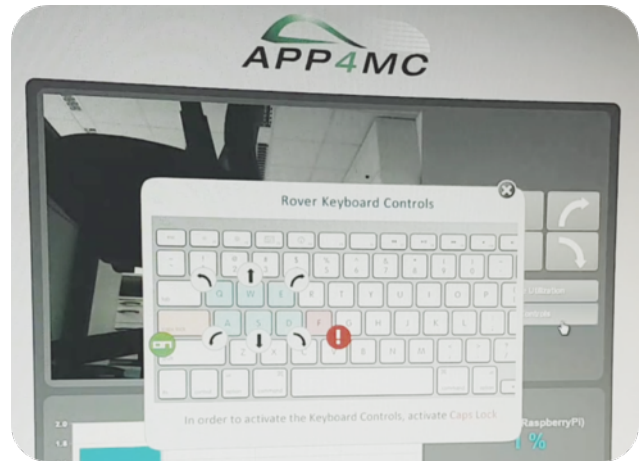
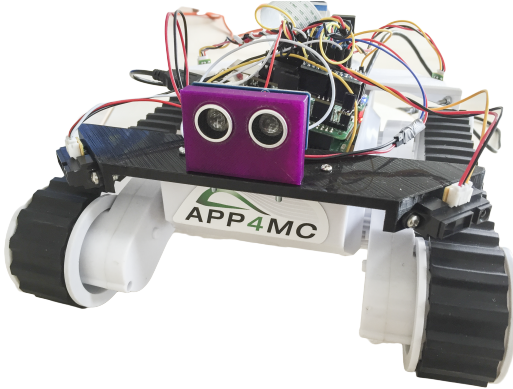
mozcelikors@gmail.com

Supervision & revision

Robert Höttger

robert.hoettger@fh-dortmund.de

University of Applied Sciences and Arts Dortmund
IDiAL Institute, Project AMALTHEA4public
BMBF Fund.Nb. 01—S14029K



1 General structure

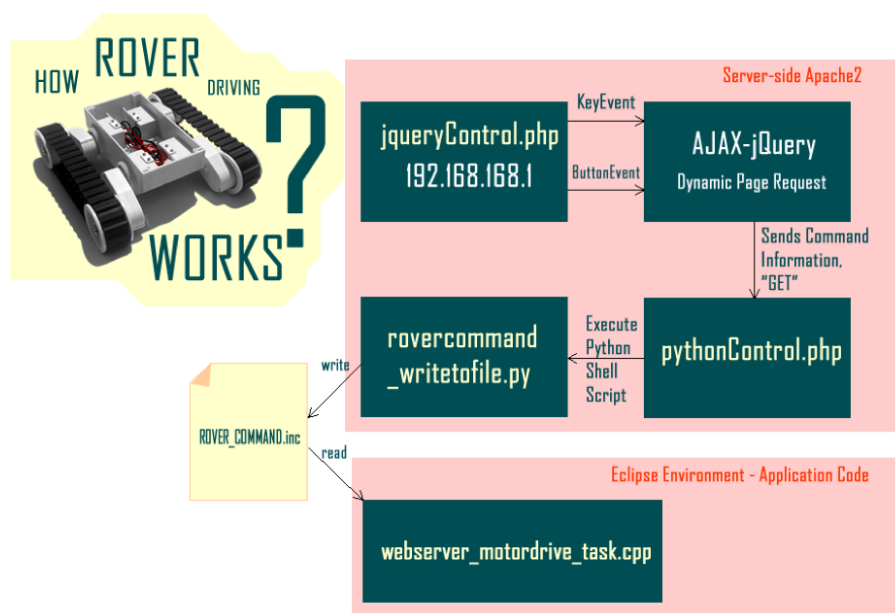


Figure 1: General structure of this documentation's rover control

2 Introduction

This documentation describes how to setup a Raspberry PI (RPI) in order to view its camera stream via a browser and further take control of the motors connected to the RPI in form of the rover. Please note that your RPI requires internet connection for some provided commands and that some of those take some time to execute.

If you work with a plain jessie (lite) distribution, in order to gain ssh access, just add an empty file named ssh to the sd card.

3 Apache2, cmake, and PHP5 Installation and Configuration

In order to start driving the rover from a web page, one should install Apache2 and PHP5. To get the latest packages from the repository list go into shell and type (*internet connectivity required*):

```
1 sudo apt-get update
2 sudo apt-get upgrade
```

To install apache2 and cmake type:

```
1 sudo apt-get install apache2 -y
2 sudo apt-get install php5 libapache2-mod-php5 -y
3 sudo apt-get install cmake
```

Afterwards, the webpage should be located at `/var/www/html` directory. In order to give certain permissions to the current user execute:

```
1 sudo chgrp -R www-data /var/www/html
2 sudo find /var/www/html -type d -exec chmod g+rx {} +
3 sudo find /var/www/html -type f -exec chmod g+r {} +
```

In the following, replace pi with your user name, which is typically "pi" in Raspbian:

```
1 sudo chown -R pi /var/www/html/
2 sudo find /var/www/html -type d -exec chmod u+rw {} +
3 sudo find /var/www/html -type f -exec chmod u+rw {} +
```

Subsequently, enable access to the Linux file-system from the provided webpage via:

```
1 sudo visudo
```

Within the editor, add the following to the end, save the file, and exit (depending on your default editor) (Saving and exiting in Nano editor `Ctrl+O → Y → Enter`, saving and exiting in Emacs editor `Ctrl+X → Ctrl+S → Enter → Ctrl+X → K → Enter`)

```
1 www-data ALL=(ALL) NOPASSWD: ALL
```

4 Setting up the webpage

Before we begin, we need some building tools and git in order to continue. Execute the following to make this work:

```
1 sudo apt-get install build-essential
2 sudo apt-get install git
```

After the RPI setup is complete, download the provided files from our repository, and make the following changes in order to merge it with you current `/var/www/html/` folder. **If you copy&paste the command, make sure there is not line break!**

```
1 sudo git clone https://gitlab.pimes.fh-dortmund.de/RPublic/RoverWeb.git
2 sudo cp -r /var/www/RoverWeb/* /var/www/
3 sudo rm -rf RoverWeb/
```

5 Installing psutil

In order to monitor core utilization, a python pip package called 'psutil' needs to be installed. First, we need to get the essential building and packaging tools for python. **If you copy&paste the command, make sure there is not line break!**

```
1 sudo apt-get install build-essential python-dev python3-dev python-pkg-
  resources python3-pkg-resources python-pip
2 cd /var/www/html/
3 sudo pip install psutil/
```

OPTIONAL: The last command, in some Linux distributions (such as Jessie Lite) created error regarding the pip version. In case you get an error, try to fix your pip by the following: (Replace 1.5.6 with the version of pip it requires)

```
1 sudo easy_install pip==1.5.6
```

Now, we are ready to test if our script is running. To move forward check if the following command results in an error. If it does not, hit Ctrl+C to exit the program and move onto the next section.

```
1 sudo python /var/www/html/record_core_usage_rpi.py
```

6 Installing Raspberry Pi Camera and Streamer

Note: The following parts are quite error prone. Please contact us if you have problems. Firstly, install the Raspberry Pi camera via:

```
1 sudo raspi-config
```

Select Advanced → Enable Camera (or '5 Interfacing Options' → 'P1 Camera' → '<Yes>' → '<Ok>'), and then save and reboot your Pi. Installation of the Streamer requires a .so specific file input_raspicam.so if you are using Raspicam instead of a webcam. This is provided with the files cloned from our git repository, but needs to be installed properly.

Before this is done, we should make sure the required packages are installed and libraries are properly configured:

```
1 sudo apt-get install libjpeg8-dev imagemagick libv4l-dev
2 sudo ln -s /usr/include/linux/videodev2.h /usr/include/linux/videodev.h
```

Afterwards, the mjpg-streamer has to be installed via (again, make sure there is no line break!):

```
1 cd /var/www/html/camerastuff/newmjpg-streamer/mjpg-streamer/mjpg-streamer-
  experimental
2 sudo make clean
3 sudo cmake .
4 sudo make
5 sudo make install
```

Subsequently, copy the necessary libraries and executables in the respective folders:

```
1 sudo cp mjpg_streamer /usr/local/bin
2 sudo cp output_http.so input_raspicam.so input_uvc.so /usr/local/lib/
```

Before testing the camera, permissions to the executables and libraries have to be set:

```
1 sudo chmod 777 /usr/local/lib/*.so
2 sudo chmod 777 /usr/local/bin/mjpg_streamer
```

Finally, the camera can be tested. With the following command (if there are no errors) the setup should be working. The following not only makes the executables know the location of the libraries but also sets up a reliable and fast input connection for raspberry pi camera:

```
1 sudo bash /var/www/html/camerastuff/webcam_stream_start_from_rpi_camera.sh
```

The following provides the specific content of this file, that can be adjusted towards specific needs:

```

1 #!/bin/bash
2 sudo bash /var/www/html/camerastuff/newmjpg-streamer/mjpg-streamer/mjpg-
  streamer-experimental
3 export LD_LIBRARY_PATH=/usr/local/lib
4 sudo /usr/local/bin/mjpg_streamer -i "/usr/local/lib/input_raspicam.so -x 640
  -y 480 -fps 30" -o "/usr/local/lib/output_http.so -w /usr/local/www -p 8081"

```

After starting this .bash file, the webpage at <http://192.168.168.1/jqueryControl.php> should show the camera stream.

7 Adjustments to the rover application

In order to update the Eclipse application that is written with pThreads, one could clone the files from the repository to their Eclipse workspace. To do that, download the files from the following Git repository and then import the sources located at [rover/src/](#) to your workspace.

```
1 https://git.eclipse.org/r/app4mc/org.eclipse.app4mc.examples
```

Basics about setting up Eclipse is given at goo.gl/vT5PF8.

8 Completion

In order to be able to use the web-page with its all functions, the following commands have to be executed on the rover before connecting to the webpage after a reboot.

```

1 sudo python /var/www/html/initialize.py
2 sudo bash /var/www/html/camerastuff/webcam_stream_start_from_rpi_camera.sh &
3 sudo python /var/www/html/record_core_usage_rpi.py &

```

To make everything permanent, it is suggested to insert some commands into `/etc/rc.local` file. Therefore, type:

```

1 cd /etc/
2 sudo nano rc.local

```

And paste the following before the `'exit 0'` command line:

```

1 cd /var/www/html/
2 sudo python /var/www/html/initialize.py &
3 sudo python /var/www/html/record_core_usage_rpi.py &
4 sudo bash /var/www/html/camerastuff/webcam_stream_start_from_rpi_camera.sh &

```

Save and close via *Ctrl*→*X*, *Y* → *Enter*.

To see the camera stream and to control the rover, use your browser and type in:

```
1 http://<Your\_IP\_Address>/jqueryControl.php
```

If your rover provides the hotspot from PolarSys, use

```
1 http://192.168.168.1/jqueryControl.php
```

9 Reporting

If you have any problems, hints, bugs, or similar, please let the authors of this document know.