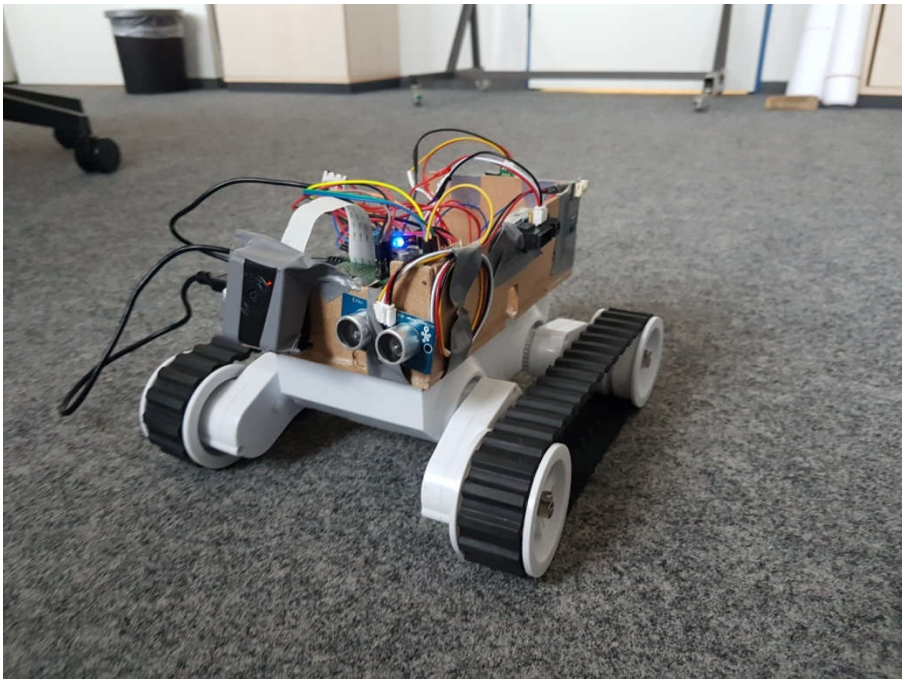
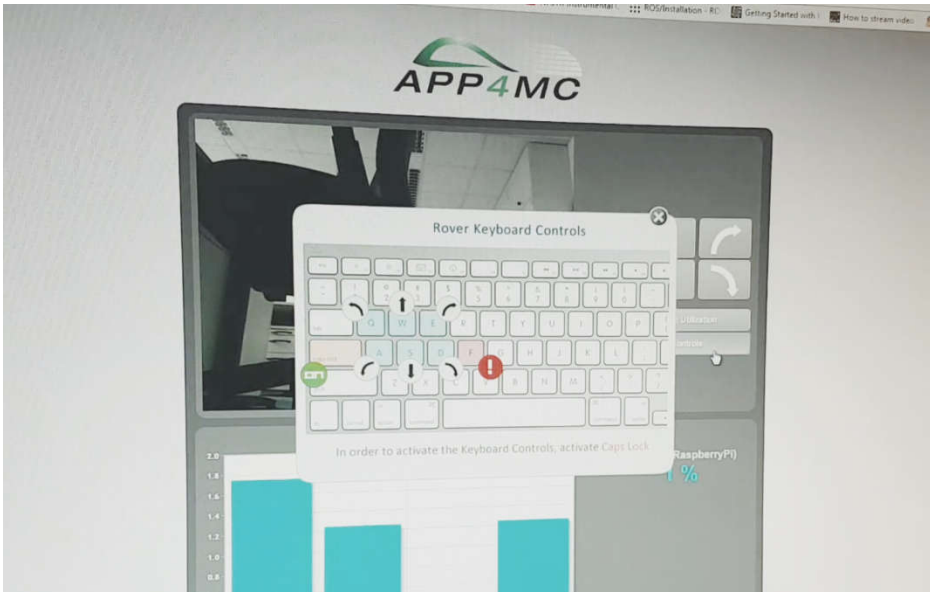
ROVER WEBPAGE-BASED DRIVING DOCUMENTATION

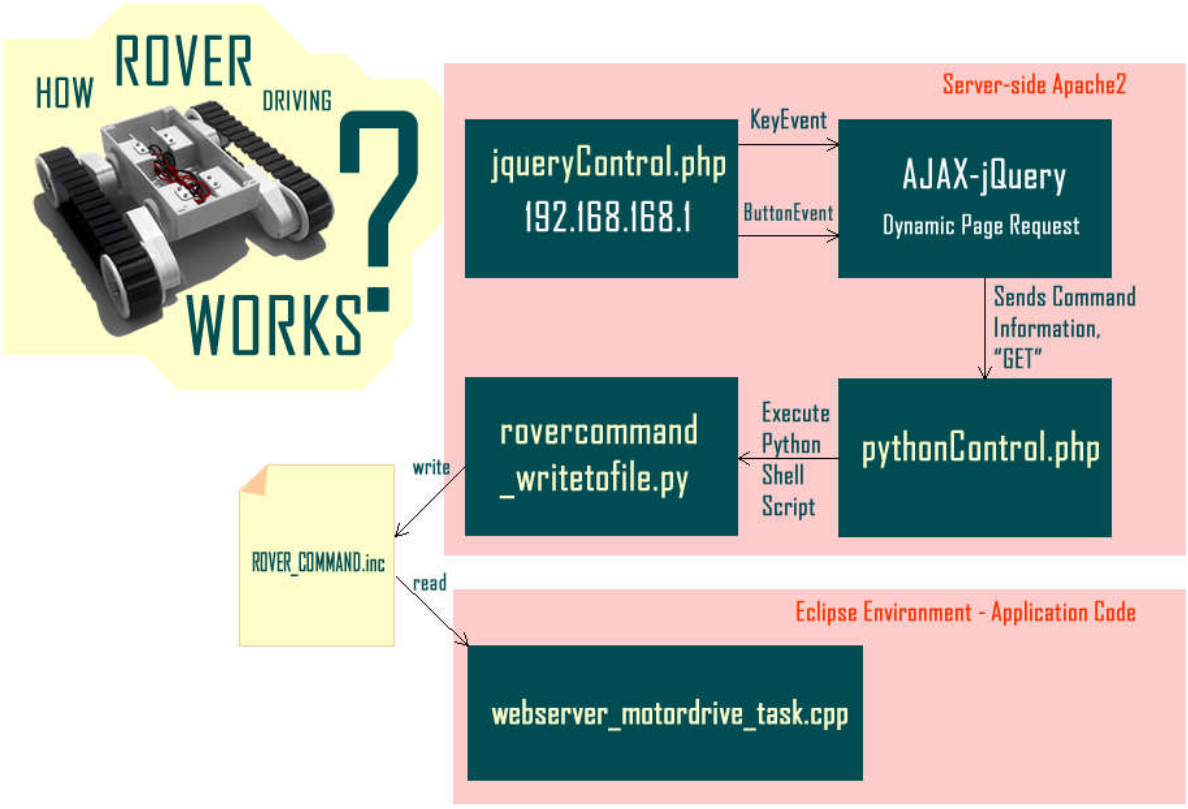




Keyboard and Button Control Webscript Guide for Rover
Rev. 002

Implementation and Documentation: M.Özcelikörs <mozcelikors@gmail.com>
Supervision and Revision: R.Höttger
FH Dortmund - IDiAL Institute

How it works?



About the script provided

The script provided does not need extra configuration to setup IP addresses, which are dynamically obtained. The only requisite for it to work is connecting to the same IP address and then going into http://<Your_IP_Address>/jqueryControl.php from your favourite browser.

If your rover has the hotspot from PolarSys, use http://192.168.168.1/jqueryControl.php

Apache2 and PHP5 Installation and Configuration

In order to start driving rover from web page, one should install Apache2 and PHP5 to begin with:

To get the newest packages repository list and upgrading,

```
sudo apt-get update

sudo apt-get upgrade
```

To install apache2;

```
sudo apt-get install apache2 -y

sudo apt-get install php5 libapache2-mod-php5 -y
```

Now you should have your web page at /var/www/html directory.
In order to give certain permissions to the current user;

```
sudo chgrp -R www-data /var/www/html

sudo find /var/www/html -type d -exec chmod g+rx {} +

sudo find /var/www/html -type f -exec chmod g+r {} +
```

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

```
sudo chown -R USER /var/www/html/

sudo find /var/www/html -type d -exec chmod u+rwx {} +

sudo find /var/www/html -type f -exec chmod u+rw {} +
```

Now, we want to enable access to Linux file-system from our web page. To do that;

```
sudo visudo
```

In the page add the following to the end of the page, Then save and exit;
(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)

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

Setting up the Web page

Now that everything has been setup, we need to download the zip file provided. After you downloaded the file, place that file into your web server folder by;

```
sudo mv /home/pi/Downloads/RoverWebpage_FHDO.zip /var/www/
```

Then, extract the archive by doing;

```
cd /var/www/
```

```
sudo unzip RoverWebpage_FHDO.zip
```

Now that you have all the files, you can delete the archive file if you like;

```
sudo rm -rf RoverWebpage_FHDO.zip
```

Installing psutil

There is a python pip package called 'psutil' which is a library to monitor cores, to use this, we should first install it.

Doing so is quite simple;

```
sudo apt-get install python-dev python3-dev
```

```
cd /var/www/html/
```

```
sudo pip install psutil/
```

Installing Raspberry Pi Camera and Streamer

There might be some errors you are getting with this section, please contact us if it's the case and we'll fix it.

In order to install Raspberry Pi camera, one should the following;

```
sudo raspi-config
```

Select Advanced > Enable Camera, 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 .zip file, but needs to be installed properly.

Now, we should first clean up the build.

```
cd /var/www/html/camerastuff\newmjpg-streamer\mjpg-streamer\mjpg-streamer-experimental
```

```
sudo make clean
```

And then clean up the CMAKE cache by;

```
sudo rm -rf CMakeCache.txt
```

Now, we are ready to install it;

```
sudo cmake .
```

```
sudo make
```

```
sudo make install
```

Copy the necessary libraries and executables in the respective folders;

```
sudo cp mjpg_streamer /usr/local/bin
```

```
sudo cp output_http.so input_raspicam.so input_uvc.so /usr/local/lib/
```

One more thing left to do before testing the camera is to give permissions to executables and libraries. To do that;

```
sudo chmod 777 /usr/local/lib/*.so
```

```
sudo chmod 777 /usr/local/bin/mjpg_streamer
```

Now we should test the camera.

After the following command, if you get no errors, your 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.

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

Here is the content of this file, if somebody wants to adjust it;

```
#!/bin/bash
sudo bash /var/www/html/camerastuff/newmjpg-streamer/mjpg-streamer/mjpg-streamer-experimental
export LD_LIBRARY_PATH=/usr/local/lib
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"
```

When you start this .bash file, you can check if your camera is working by going to http://192.168.168.1 from your browser.

How to Run Everything

In order to be able to use the web-page with its all functions, you should enter the following commands in Rover before connecting to webpage after a reboot.

```
sudo python /var/www/html/initialize.py &

sudo bash /var/www/html/camerastuff/
webcam_stream_start_from_rpi_camera.sh  &

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

To make everything permanent, one should consider putting what's written here inside **/etc/rc.local** file.

Before `exit 0` command in /etc/rc.local/;

Add

```
cd /var/www/html/

sudo python /var/www/html/initialize.py &

sudo python /var/www/html/record_core_usage_rpi.py &

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

Control-side Application Adjustment

You should also adjust your rover control application code in order to drive motors according to the input from the webpage. To do this, find the newest version of the Eclipse application, which contains necessary adjustments to include following task function:

```
void *WebServer_MotorDrive_Task(void * arg)
{
	FILE *fp;
	char ch;

	while(1)
	{
```

```
            fp = fopen("/var/www/html/ROVER_CMD.inc", "r");
            ch = fgetc (fp);
            //printf("Got command = %c\n", ch);
            pthread_mutex_lock(&keycommand_lock);
            keycommand_shared = tolower(ch);
            pthread_mutex_unlock(&keycommand_lock);
            fclose(fp);
            delayMicroseconds(50000);//50ms
        }
}
```