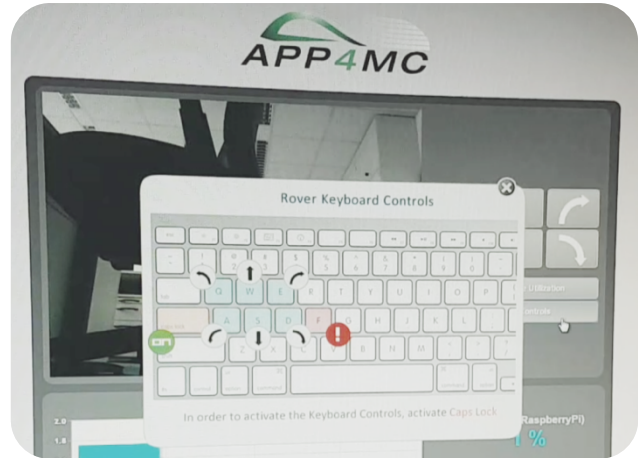
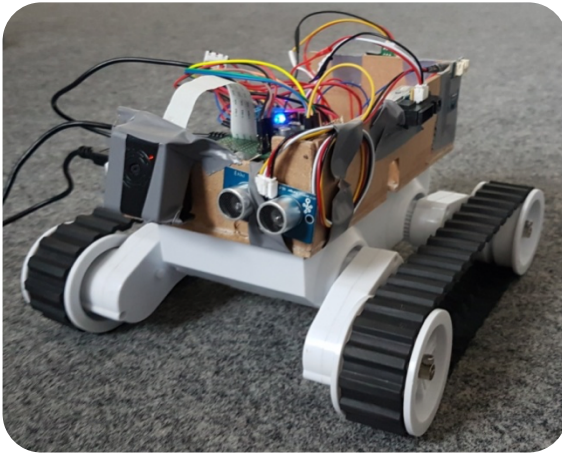


ROVER DOCUMENTATION

Browser-based camera stream & drive control

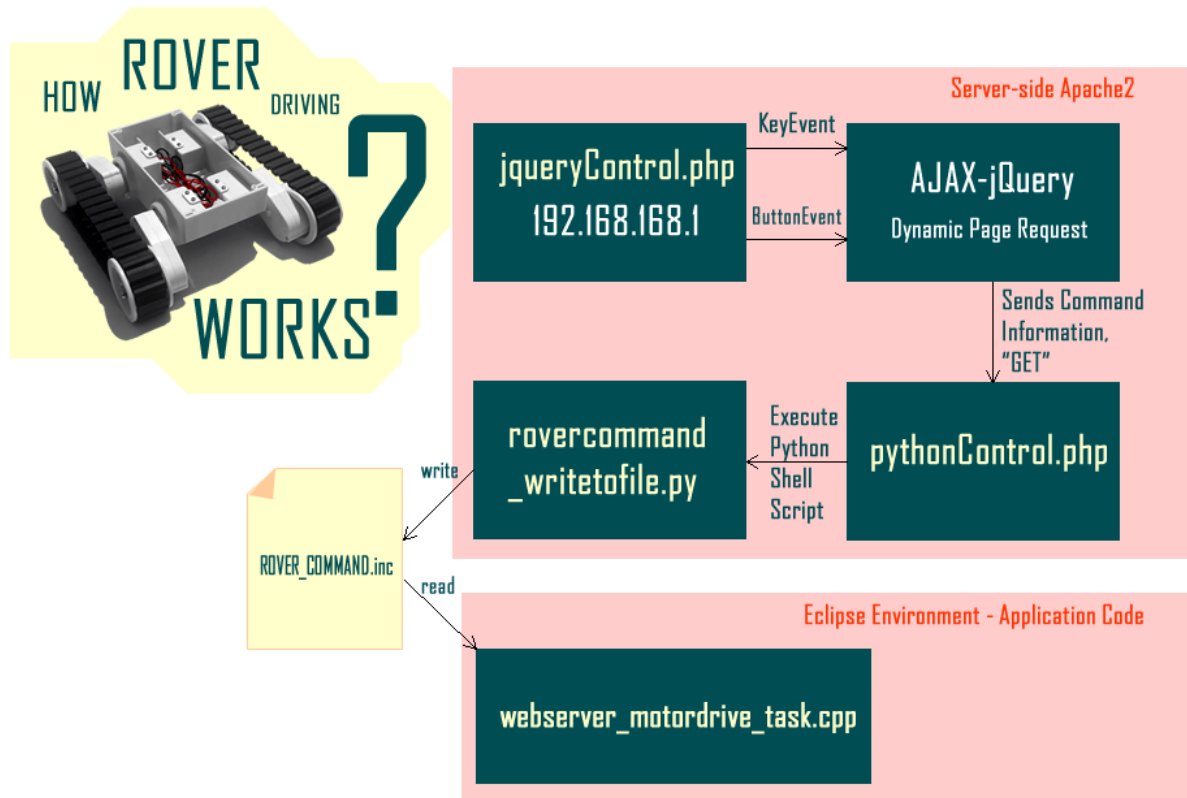


Keyboard and Button Control Webscript Guide for Rover (APP4MC)

Implementation and Documentation:
Supervision and Revision:

M. Özcelikörs <mozcelikors@gmail.com>
R. Höttger <Robert.hoettger@fh-dortmund.de>
FH Dortmund - IDiAL Institute – APP4MC rover

Structure:



About this script

The script provided does not require extra configuration to setup IP addresses, which are dynamically obtained. The only requisite is connecting to the same IP address via the following in your browser:

`http://<Your_IP_Address>/jqueryControl.php`

If your rover provides the hotspot from PolarSys, use

`http://192.168.168.1/jqueryControl.php`

Apache2 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*):

```
sudo apt-get update
sudo apt-get upgrade#
```

To install apache2 type:

```
sudo apt-get install apache2 -y
sudo apt-get install php5 libapache2-mod-php5 -y
```

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

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

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

```
sudo visudo
```

Within the opened page, add the following to the end, save the file, 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

After the RPI setup is complete, move the provided webpage zip file to the web server folder, extract it, and delete the zip via:

```
sudo mv /home/pi/Downloads/RoverWebpage_FHDO.zip /var/www/
cd /var/www/
sudo unzip RoverWebpage_FHDO.zip
sudo rm -rf RoverWebpage_FHDO.zip
```

Installing psutil

In order to monitor core utilization, a python pip package called 'psutil' needs to be installed:

```
sudo apt-get install python-dev python3-dev
cd /var/www/html/
sudo pip install psutil/
```

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:

```
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
```

Afterwards, clean up the CMAKE cache and install via:

```
sudo rm -rf CMakeCache.txt
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:

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

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

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

```
#!/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"
```

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

How to Run Everything

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.

```
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, it is suggested to put the following into a **/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

Since the rover does not yet consider the webpage commands, the rover control application code must be edited. To do so, find the newest version of the Eclipse application, which contains necessary adjustments and include the 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
    }
}
```