Parts:

* Raspberry Pi 3
* L293D motor controller
* Powerbank
* Pi camera module
* Motors

Basic description:

Flask Controller is a video streamer with added buttons to send post requests

In test.py there is one inefficiency as it writes and reads to sd card repeatedly, this is fixed in test2.py

Pre-requisites:

OpenCV installed( i have tested on opencv 4)

Need to be run in opencv virtualenv( source ~/.profile followed by workon cv)

Enable i2c and camera in rpi interfacing settings

Installation:

Tutorial:

<https://www.hackster.io/ruchir1674/video-streaming-on-flask-server-using-rpi-ef3d75>

IMPORTANT: On EVERY BOOT and after installing all packages from above link, do the following

This is to enable the video driver- modprobe bcm2835-v4l2 driver

We can make the pi do this on boot by editing the

**shuckle wrote:**Just add the command to /etc/rc.local file.

That's not the 'standard' way of loading a module.

You should add it to /etc/modules. For example using:

**Code: [Select all](https://www.raspberrypi.org/forums/viewtopic.php?t=108038#)**

sudo nano /etc/modules

Add a new line containing:

**Code: [Select all](https://www.raspberrypi.org/forums/viewtopic.php?t=108038#)**

bcm2835-v4l

Then save, exit and reboot

Command: sudo modprobe bcm2835-v4l2

Possible errors when driver not run errors :

a)In case of empty jpeg error

OR

b)VIDEOIO ERROR: V4L: can't open camera by index 0

Troubleshooting:

* If the page does not load, terminal shows that program is stuck at stat statement, then set debug to false in the python flask program’s app.run statement
* Most camera problems can be solved by sudo modprobe statement
* In addition we can check if uv4l is working by going to the <ipaddr>:8080 in a browser, it should show a page with multiple options in a GUI including streaming, if on clicking streaming we can see the live-stream then it mostly means we need to execute modprobe statement or something else
* Do not run the program as sudo from virtualenv

FlaskController:

The test.py program establishes a flask server serving the html page with the livestream and buttons

Each time a button is pressed a corresponding function from another python program(example: forward.py) is called to perform motor movement

For connections to the motor:

* Connect motor -ve to RPi gnd
* Connect motor +ve to RPi GPIO pin with pwm (Ex: GPIO 21)
* Connect controller EN to any GPIO general
* Connect the motor controller to external power source

FlaskController2:

Improvement on FlaskController by using XMLHttp, this prevents refresh of page everytime a button is clicked, take care to ensure reqeust.form[] values and keys are same in the html code and python code

Referred to this code to write:<https://gist.github.com/KentaYamada/2eed4af1f6b2adac5cc7c9063acf8720>

FlaskController3:

Added buttons by sending the button id in the request back to flask, removed useless prints, cleaned up code, added py scripts for 4 direction motion using pwm

PWM is applied on the enable pin and the polarity of the input pins determines the direction of movement for each motor

Check any one of the direction py files to see which GPIO pins are connected to the motor controller