Table of Contents (Clickable links!)

Flashing the Raspberry Pi for mjpg_streamer

As of July 30, 2022, the mjpg_streamer version we use is still not compatible with the libcamera-vid functionalities of the Bullseye Raspbian system. However, there is hope with this new branch: https://github.com/ArduCAM/mjpg-streamer

The workaround I currently use is:

1. Download **2020-02-14 Raspbian version** (which works with mjpg_streamer)
2. Once that is downloaded to your computer, use BalenaEtcher to flash your SD card.

Updating the Raspbian software

At any rate, with this flashing Raspberry PI workaround, we have to keep in mind that this is an "oldstable" version of Raspbian. The most important thing is to run the following before you install any of the necessary packages:

```
sudo apt-get update --allow-releaseinfo-change
```

This allows you to update your raspberry pi even though we are using Bullet, which is an "oldstable" version. Reference: https://forums.raspberrypi.com/viewtopic.php?t=318302

Enabling camera and ssh

In a terminal type the following:

```
sudo raspi-config
```

A list menu will appear. Choose the following:

- 5 Interfacing Options
  - P1 Camera → enable
- 5 Interfacing Options
  - P2 SSH → enable
  - Note: enabling ssh allows you to

This requires `sudo reboot`, but we can wait until after you set up the static IP address.

Setting up static IP address

To set up a static IP address on the Raspberry Pi, you must edit the /etc/dhcpcd.conf file. Type the following in a terminal:

```
nano /etc/dhcpcd.conf
```

1. Find the commented out line that says `# Example static IP configuration`:
2. Do not uncomment anything but type the following four lines below:

```
interface eth0
static ip_address=[x1.x2.x3.positive_int_that_is_not_1_and_smaller_than_255]/24
static routers=[x1.x2.x3.1]
static domain_name_servers=[x1.x2.x3.1] 8.8.8.8
```

3. Don't forget to sudo reboot (although that can also wait until after setting up your Raspberry Pi so it can automatically run mjpg_streamer).
4. TODO: add how to set up connection on Windows/Linux (I think Apple is a lot easier to set it up anyways)

Downloading mjpg-streamer

Git clone the repo and follow its README instructions:
https://github.com/jacksonliam/mjpg-streamer

Note: in the README, it says that you should go directly to the mjpg-streamer-experimental folder from the Home directory. This is incorrect, you must first cd into the mjpg-streamer folder. If you want to test the download, do the following:

1. Make sure you have rebooted the Raspberry Pi if you have not done so already after enabling camera and ssh.
2. Follow the instructions in the Testing mjpg_streamer section.

Testing mjpg_streamer

1. In the terminal, type the following:

```
mjpg_streamer -o "output_http.so -w ./www" -i "input_raspicam.so -fps 30 -x 640 -y 480"
```

2. To see a picture, use the web browser and type:
   http://[ip_address]:8080/?action=stream

   Note: use the static IP address that was set up previously.

Automatically running mjpg_streamer command on Raspberry Pi

1. In the terminal, type the following:

```
nano ~/.bashrc
```

2. Add the following line to the end of the file (this is the same as the one used here):

```
mjpg_streamer -o "output_http.so -w ./www" -i "input_raspicam.so -fps 30 -x 640 -y 480"
```

3. To make sure it worked, close the terminal and reopen a new one. Feel free to check the web browser again using 2. of the Testing mjpg_streamer section.
4. To interface with a computer, you will need a script running on the computer.

GelSight interface with connected computer

Note that this code only works once the computer is properly set up to interface with the Raspberry Pi (see the Setting up static IP address section).

Streaming video code (example code)

```
import cv2
import numpy as np
import time
import urllib.request

from threading import Thread

img = None
def start_stream():
    global img
    stream = urllib.request.urlopen('http://10.0.12.12:8080/?action=stream')
#Raspberry pi camera url (mjpg_streamer)
    bytes = b''
    while True:
        bytes += stream.read(1024)
        a = bytes.find(b'\xff\xd8')
        b = bytes.find(b'\xff\xd9')
        if a != -1 and b != -1:
            jpg = bytes[a:b+2]
            bytes = bytes[b+2:]
            img = cv2.imdecode(np.frombuffer(jpg, dtype=np.uint8),
cv2.IMREAD_COLOR)

thread = Thread(target=start_stream)
thread.daemon = True
thread.start()
time.sleep(1)

while(1):
    try:
        cv2.imshow('frame', img)
        if cv2.waitKey(25) & 0xFF == 27:
            break

    except Exception:
        break
cv2.destroyAllWindows()
```

For a code breakdown, please refer to this document (GelSight “Interface” with ROS).