

Setup Raspy with OpenCV

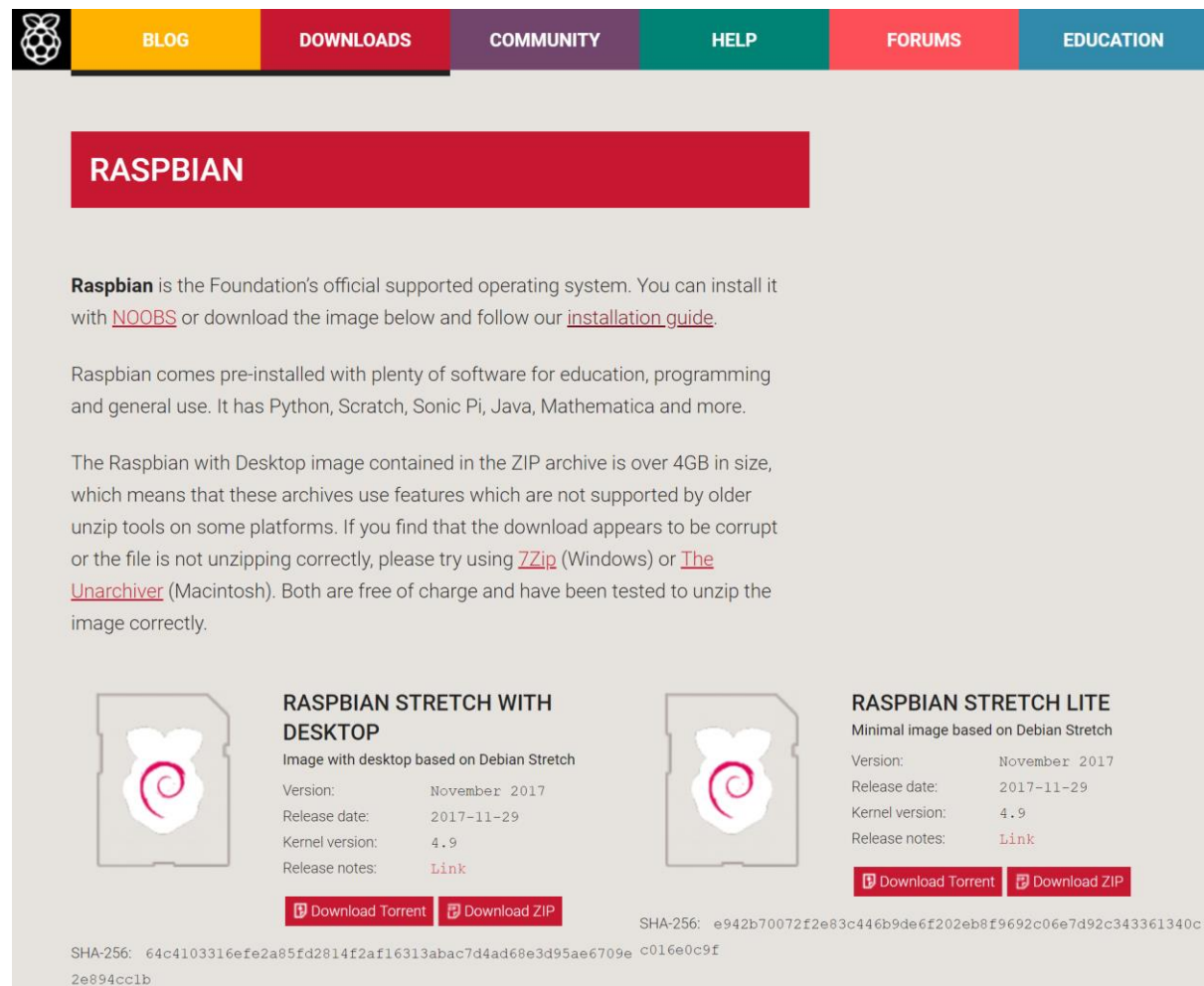
From a Dummy for other Dummies

Install Raspbian

Download the latest image from raspberrypi.org. I took the lite version without desktop gui.

To check the hashcode on Windows you can use the following command:

```
certutil -hashfile C:\Users\...raspbian-stretch-lite.zip sha256
```



The screenshot shows the Raspbian download page. At the top is a navigation bar with links: BLOG, DOWNLOADS, COMMUNITY, HELP, FORUMS, and EDUCATION. Below this is a red header with the word "RASPBIAN". The main content area has a paragraph about Raspbian being the official supported operating system, with links to "NOOBS" and an "installation guide". It also mentions pre-installed software like Python, Scratch, Sonic Pi, Java, and Mathematica. A warning paragraph states that the desktop image is over 4GB and may not unzip correctly on older tools, recommending 7Zip or The Unarchiver. At the bottom, there are two download options: "RASPBIAN STRETCH WITH DESKTOP" and "RASPBIAN STRETCH LITE". Each option includes a version number (November 2017), release date (2017-11-29), kernel version (4.9), and a link to release notes. Below these are buttons for "Download Torrent" and "Download ZIP". A SHA-256 hash is provided at the bottom: e942b70072f2e83c446b9de6f202eb8f9692c06e7d92c343361340c2e894cc1b.

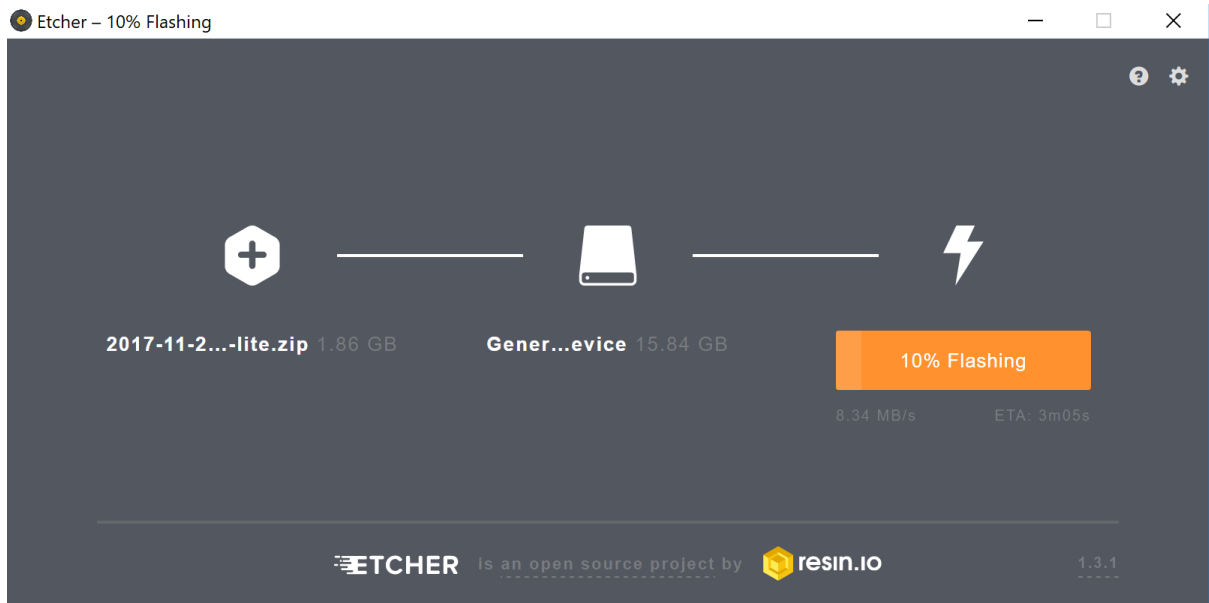
Image with desktop based on Debian Stretch	Minimal image based on Debian Stretch
Version: November 2017	Version: November 2017
Release date: 2017-11-29	Release date: 2017-11-29
Kernel version: 4.9	Kernel version: 4.9
Release notes: Link	Release notes: Link

SHA-256: e942b70072f2e83c446b9de6f202eb8f9692c06e7d92c343361340c2e894cc1b

Link: <https://www.raspberrypi.org/downloads/raspbian/>

Write the image to the SD card either manually or with a client like Etcher. I used Etcher.

Select the previously downloaded image (.zip File), then select the SD Card and click the flash button in Etcher.



Etcher Link: <https://etcher.io/>

Installation Guide Link: <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

Setup wireless connection

You could work on your Raspy with a HDMI Monitor and a USB Keyboard locally. I want to work via ssh over the wlan hotspot of my smartphone. To make the Raspy connect to the network we have to configure the ssid and the password in `/etc/wpa_supplicant/wpa_supplicant.conf` as follows:

```
country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="<your ssid>"
    psk="<your password>"
}
```

After configuring this, we have to restart the wlan client:

```
wpa_cli -i wlan0 reconfigure
```

Basic configuration

Now the raspy should connect to the network. The next point is to expand the root filesystem and to start the ssh server. Raspbian brings a shell client for these options:

```
sudo raspi-config
```

Download OpenCV and setup python environment

```
wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.3.0.zip
```

```
unzip opencv.zip
```

```
wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.3.0.zip
```

```
unzip opencv_contrib.zip
```

```
wget https://bootstrap.pypa.io/get-pip.py
```

```
python3 get-pip.py
```

```
pip install virtualenv virtualenvwrapper
```

```
mkvirtualenv cv -p python3
```

```
workon cv
```

```
pip install numpy
```

Compile OpenCV

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D  
INSTALL_PYTHON_EXAMPLES=ON -D OPENCV_EXTRA_MODULES_PATH=~/.opencv_contrib-  
3.3.0/modules -D BUILD_EXAMPLES=ON ..
```

```
-- Python 3:
```

```
-- Interpreter:      /home/pi/.virtualenvs/cv/bin/python3 (ver 3.5.3)
```

```
-- Libraries:       /usr/lib/arm-linux-gnueabi/libpython3.5m.so (ver 3.5.3)
```

```
-- numpy:           /home/pi/.virtualenvs/cv/lib/python3.5/site-packages/numpy/core/include  
(ver 1.14.1)
```

```
-- packages path:    lib/python3.5/site-packages
```

```
python -m site
```

```
sys.path = [
```

```
    '/home/pi/opencv-3.3.0/build',
```

```
    '/home/pi/.virtualenvs/cv/lib/python35.zip',
```

```
    '/home/pi/.virtualenvs/cv/lib/python3.5',
```

```
    '/home/pi/.virtualenvs/cv/lib/python3.5/plat-arm-linux-gnueabi',
```

```
'/home/pi/.virtualenvs/cv/lib/python3.5/lib-dynload',  
'/usr/lib/python3.5',  
'/usr/lib/python3.5/plat-arm-linux-gnueabi',  
'/home/pi/.virtualenvs/cv/lib/python3.5/site-packages',  
]  
USER_BASE: '/home/pi/.local' (doesn't exist)  
USER_SITE: '/home/pi/.local/lib/python3.5/site-packages' (doesn't exist)  
ENABLE_USER_SITE: False
```

Change swapfile to 1024

```
sudo /etc/init.d/dphys-swapfile stop  
sudo /etc/init.d/dphys-swapfile start
```

```
make -j4  
sudo make install  
sudo ldconfig
```

Change swapfile back to 100

<https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/>

Setup git

```
sudo apt-get install git  
git config --global user.name "Raspberry Pi"  
git config --global core.editor vi  
  
ssh-keygen -t rsa -C pi@raspy  
VIELLEICHT: sudo /etc/init.d/ssh restart  
git clone git@github.com:haubschueh/loufchatz.git loufchatz  
  
sudo mkdir /opt/loufchatz  
sudo chown pi /opt/loufchatz/
```

```
git clone git@github.com:haubschueh/loufchatz.git loufchatz
```

```
cd loufchatz/
```

```
git pull
```

Serial interface

VIELLEICHT: Check if serial interface is used by the operating system

```
dmesg | grep tty
```

Disable the serial interface over the the console gui:

```
sudo raspi-config
```

Install the package pyserial with pip. It's important to install pyserial and not serial, otherwise it might not be possible to open ports.

```
pip install pyserial
```

```
python3
```

```
>> import serial
```

```
>> port = serial.Serial()
```

Disable Bluetooth modem that blocks GPIO ports

Add to the end of the file /boot/config.txt

```
dtoverlay=pi3-disable-bt
```

We also need to run to stop BT modem trying to use UART

```
sudo systemctl disable hciuart
```

```
dtc -I fs /proc/device-tree
```

```
uart1_pins {  
    brcm,pins;  
    phandle = <0x11>;  
    brcm,pull;  
    brcm,function;  
};
```

[...]

```
uart0_pins {  
    brcm,pins = <0x20 0x21>;  
    phandle = <0x8>;  
    brcm,pull = <0x0 0x2>;  
    brcm,function = <0x7>;  
};
```