

Download and flash Gumstix's Raspbian Lite image

Download the Gumstix Raspbian Lite image from:

<https://gumstix-raspbian.s3.amazonaws.com/2020-01-12/raspberrypi-cm3/rpi-4.19.y/2019-09-26-raspbian-buster-lite.img.xz>

Flash your SD card with the extracted image above. To do this, open a terminal on your host computer.

```
$ sudo dd if=2019-09-26-raspbian-buster-lite.img of=/dev/sdX bs=4MiB
```

Where sdX is the location of your SD card (to find the location of the SD card, you can use the command "lsblk" to match the memory device with the SD card).

Setting up SSH:

Mount the 2 partitions (boot and rootfs) of the SD card.

- 1) In the boot partition, create an empty file called "ssh"
- 2) In the rootfs partition, navigate to /etc/wpa_supplicant/ and open the file "wpa_supplicant.conf".
 - a) Copy and paste the following lines into wpa_supplicant.conf:

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

network={
  ssid="WIFI_SSID"
  scan_ssid=1
  psk="WIFI_PASSWORD"
  key_mgmt=WPA-PSK
}
```

Enter your WIFI_SSID and WIFI_PASSWORD for the ssid and psk respectively. You will also need to fill in the entry for "country" with the ISO code of your country (eg: Canada = CA). Note, the file will already contain the lines "ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev" and

“update_config=1”, so you can add to the file so that its contents matches what is shown above, or just overwrite the entire file with the lines above.

- 3) In the rootfs, navigate to /etc/ssh/sshd_config. Edit the sshd_config file, by adding the following line to the end of the file:

```
IPQoS cs0 cs0
```

If you have trouble saving the sshd_config file, try running “sudo su”, and then try editing and saving the sshd_config file.

Insert the SD card into your Raspberry Pi Zero. Now your Raspberry Pi Zero should be accessible via SSH

Power up your Raspberry Pi Zero and give it a minute to boot up. Then, in a terminal on the host computer, enter the following command:

```
$ ssh pi@raspberrypi.local
```

Now you should have SSH access to the Raspberry Pi Zero.

Enable the Camera

```
$ sudo raspi-config
```

Navigate to “Interfacing Options”, and enable the Camera. The Raspberry Pi Zero will ask you if you want to reboot after this, select “Yes”. It’ll end the ssh session, and you’ll have to again ssh into the Raspberry Pi Zero by running: “ssh pi@raspberrypi.local” in a terminal.

Update your Raspberry Pi Zero

```
$ sudo apt update
```

Install Gstreamer1.0

```
$ sudo apt-get install gstreamer1.0-tools \  
    gstreamer1.0-plugins-bad \  
    gstreamer1.0-plugins-good \  
    gstreamer1.0-omx \  
    python3-gi \  
    python-gst-1.0
```

If at any time you get an error that says “Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?” or “Failed to fetch...” when running the gstreamer1.0 install commands, run “sudo apt-get update --fix-missing”, and try running the failing command again.

Install rpicasrc

```
$ sudo apt-get install git
```

```
$ git clone https://github.com/thaytan/gst-rpicamsrc.git
```

```
$ sudo apt-get install autoconf \  
    libtool \  
    libgstreamer1.0-dev \  
    libgstreamer-plugins-base1.0-dev
```

```
$ cd gst-rpicamsrc
```

```
$ ./autogen.sh
```

```
$ make
```

```
$ sudo make install
```

If at any time you get an error that says “Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?” or “Failed to fetch...” when running the “sudo apt-get install

autoconf libtool libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev” command, run “sudo apt-get update --fix-missing”, and try running the failing command again.

Copy the required files over to the SD card

At this point, your Raspberry Pi Zero has all the necessary software to run this demo. Turn off the Raspberry Pi Zero, and insert the SD card into the host computer.

Mount the 2 partitions (boot and rootfs) of the SD card.

- 1) Download a copy of the **bmi160-i2c.dtbo**, and copy the bmi160-i2c.dtbo into the overlay folder, within the boot partition (ie: /boot/overlays/)
- 2) Open the config.txt file in the boot partition, and enable the bmi160-i2c.dtbo overlay by adding the following line into the config.txt:

```
dtoverlay=bmi160-i2c
```

- 3) Download a copy of **sensor_overlay.py**
 - a) Open sensor_overlay.py so that you can edit it's contents
 - b) On line 44, change the <HOST_IP_ADDRESS> to the IP address of your host machine, to which you will be streaming the video to
- 4) Copy the sensor_overlay.py into the rootfs/home/pi folder

Remove the SD card from your host computer and place it back onto the Raspberry Pi Zero. Power on the Raspberry Pi Zero and ssh into it again.

Stream Video from your Raspberry Pi Zero to your Host Computer

On your Raspberry Pi Zero, run the following command to begin streaming video data to your host machine:

```
$ python3 sensor_overlay.py /sys/bus/iio/devices/iio\:device0/
```

On your host machine, run the following command in a terminal:

```
$ gst-launch-1.0 -e -v udpsrc port=5000 ! application/x-rtp, payload=96 \
! rtppjitterbuffer ! rtph264depay ! avdec_h264 \
! fpsdisplaysink sync=false text-overlay=false
```

If everything is good, a window should pop up showing the streaming video.

Explanation of the “Horizon” measurement

The “Horizon” measurement is really the angle of the vector with respect to the z-axis.

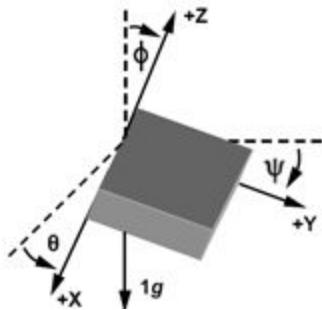


Figure 1: The Horizon angle measurement corresponds to the ϕ angle

We were able to observe ~1.5 hours of video streaming over WiFi using battery power.