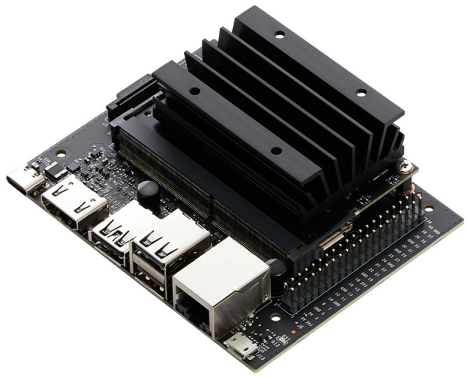
Nvidia Jetson Nano Setup

Quick installation (work in progres…)

1. Jetson Nano identification

There are two Jetson Nano models depending on the RAM capacity.

- **Jetson Nano 2 GB**: it has three USB 2.0 ports:



- **Jetson Nano 4 GB**: it has four USB 3.0 ports:

2. Download the image into a SD Card of 32 GB depending on the Jetson Nano model:

   Jetson Nano 4GB: jetson_nano_zed_2gb_image.img.gz
   Jetson Nano 2GB: jetson_nano_zed_4gb_image.img.gz

3. Get the name of the SD Card device.

```
sudo parted -l
```

```
Model: SD SD32G (sd/mmc)
Disk /dev/mmcblk0: 31.9GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 2      1049kB  1180kB  131kB                TBC
 3      2097kB  2556kB  459kB                RP1
 4      3146kB  3736kB  590kB                EBT
 5      4194kB  4260kB  65.5kB  fat32        WB0
 6      5243kB  5439kB  197kB                BPF
 7      6291kB  6685kB  393kB                BPF-DTB
 8      7340kB  7406kB  65.5kB               FX
 9      8389kB  8847kB  459kB                TOS
10      9437kB  9896kB  459kB                DTB
11      10.5MB  11.3MB  786kB                LNX
12      11.5MB  11.6MB  65.5kB               EKS
13      12.6MB  12.8MB  197kB                BMP
14      13.6MB  13.8MB  131kB                RP4
 1      14.7MB  30.4GB  30.4GB  ext4         APP
```

In this case is /dev/mmcblk0

4. Burn the image on the SD Card

```
cd ~/Download
sudo su
```

For Jetson Nano 2 GB

```
gunzip -c jetson_nano_zed_2gb_image.img.gz | dd of=/dev/mmcblk0 bs=64K status=progress
```

For Jetson Nano 4 GB

```
gunzip -c jetson_nano_zed_4gb_image.img.gz | dd of=/dev/mmcblk0 bs=64K status=progress
```

5. Jetson Nano in the robot

Connect the Jetson Nano to the internal power supply of the robot. Then connect an ethernet cable to the switch, router or PC. **From the robot PC**, check the Jetson Nano connection:

```
ping 192.168.0.50
```

Enable ssh without login:

```
ssh-keygen -t rsa
ssh-copy-id jetson@192.168.0.50
```

Press enter when it asks for the file, passphrase, same passphrase.

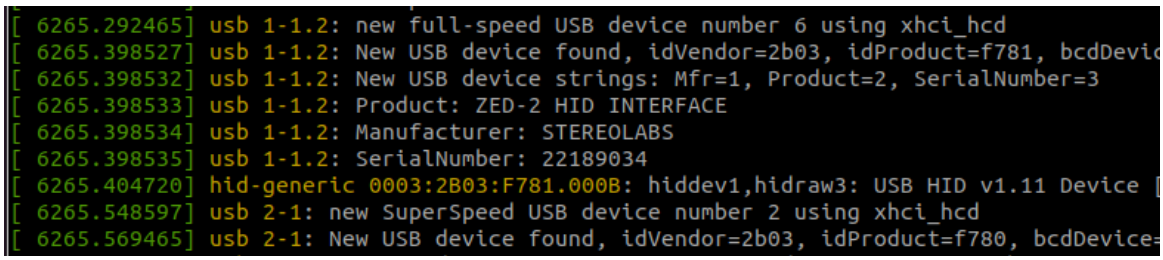
6. ZED Camera configuration:

Internet access is required the first time that a new camera is connected to a Jetson Nano. Launch the ZED camera in order enable it: zed.launch, zed2.launch or zed2i.launch

```
roslaunch zed_wrapper zed.launch
```

Finally, get the serial number of the camera. It can be obtained from the dmesg command. When the camera is connected, it will show a message like this:

```
dmesg -w
```



Create an image from an SD Card

Get the name of the device with *sudo parted -l* . In this case is /dev/mmcblk0. Then create the image:

```
sudo umount /dev/mmcblk0
```

```
sudo dd if=/dev/mmcblk0 conv=sync,noerror bs=64K status=progress | gzip -c >
jetson_nano_zed_4gb_image.img.gz
```

https://jetsonhacks.com/2020/08/08/clone-sd-card-jetson-nano-and-xavier-nx/

https://linuxize.com/post/how-to-format-usb-sd-card-linux/

Manual installation

1. Requirements

- Jetson Nano 2Gb or Jetson Nano 4Gb
- SD Card 32 GB
- 4A power supply
- Ethernet connection for the Internet
- Mouse, keyboard and monitor

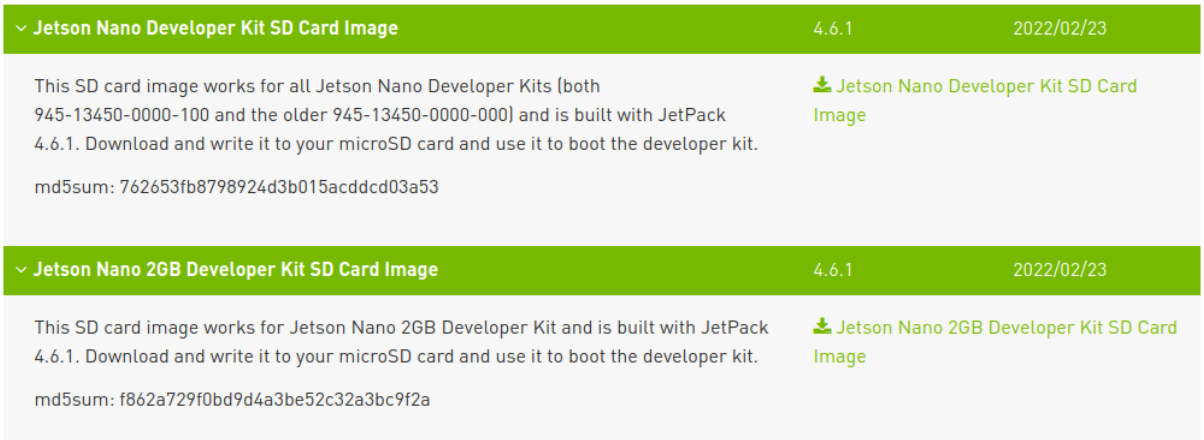
Important: Jetson Nano 2 Gb can only manage one ZED camera. Jetson Nano 4GB can work with two ZED cameras but with a frequency of 15 Hz. Be aware of the RAM model if you use two cameras.

2. SD image flashing

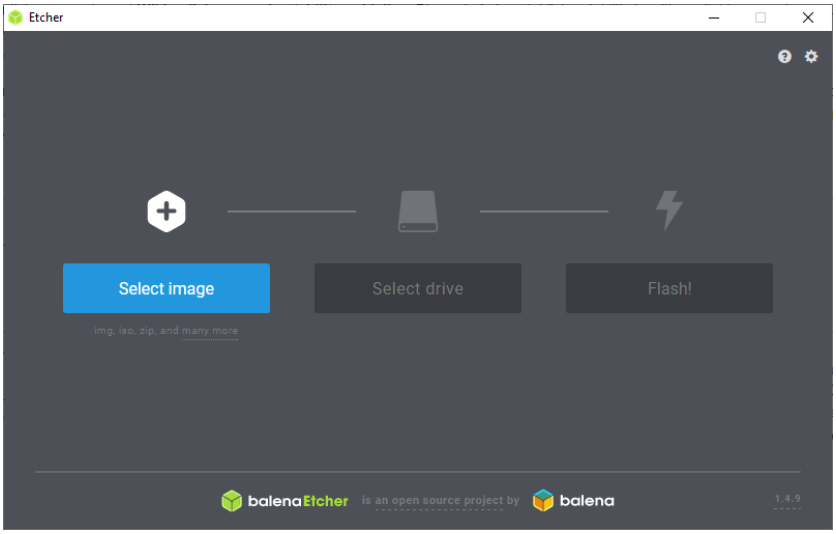
In an external machine download and flash the Jetson Nano image:

https://developer.nvidia.com/embedded/downloads#?search=SD%20card%20image

Select the SD Card image depending on the RAM memory of the Jetson Nano. You can check the Jetson Nano model here. Jetson Nano 2Gb has three USB ports, whileas, Jetson Nano 4Gb has four USB ports.

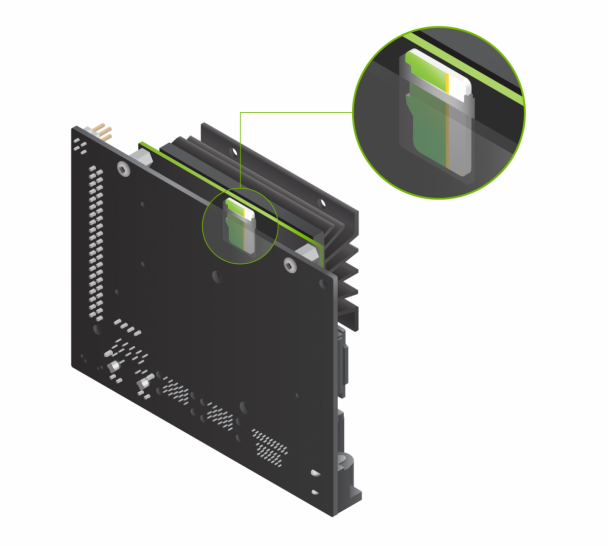


Flash the image into the SD Card. You can use the Balena Etcher program. Download and unzip the program. Open it by clicking on the AppImage icon, then, flash the SD Card:



3. First boot

Insert the SD Card into the Jetson Nano. Connect a mouse, keyboard and monitor and follow the installation steps. In addition, connect an Ethernet cable with the Internet.



This is the configuration for the system:

Name: jetson
Computer name: jetson-robotnik
Username: jetson

Password: R0b0tn1K
Autologin: enabled
Create SWAP
Maximum performance

4. ROS installation

Jetson Nano image uses Ubuntu 18.04. Follow the ROS Melodic instructions in order to set up ROS. Install the basic installation **Desktop install:**

http://wiki.ros.org/melodic/Installation/Ubuntu

5. ZED SDK installation

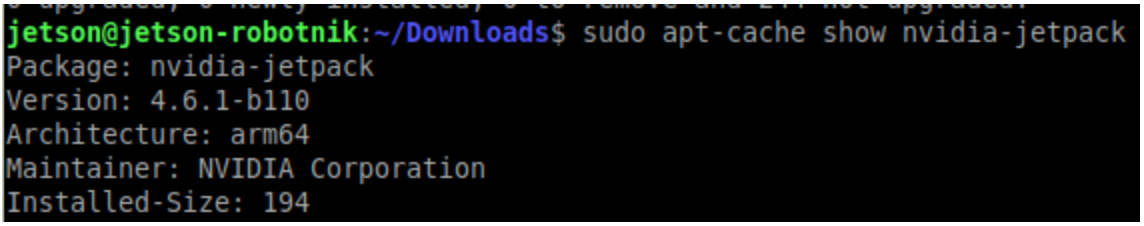
Download the ZED SDK:

https://www.stereolabs.com/developers/release/

The SDK depends on the JetPack version of the Jetson Nano (pre-installed on the image). You can check the JetPack version with this command:

```
sudo apt-cache show nvidia-jetpack
```

It returns:



In this case, we will need the ZED SDK for Jetpack 4.6.1. If the version does not match with you Jetson Nano jetpack, install the latest available version.

Install the ZED SDK. Go to the folder where the SDK was downloaded:

```
cd Downloads
```

Add permissions:

```
chmod +x ZED_SDK_Tegra_JP46_v3.7.1.run
```

Run the installer:

```
./ZED_SDK_Tegra_JP44_v3.7.1.run
```

Set YES for all the iinstallations. On Jetson Nano 2GB the Samples and API installation does not work due to a numpy dependency error.

6. ZED ROS installation

Create the catkin_ws workspace and clone the ZED ROS package:

```
cd && mkdir -p catkin_ws/src
cd ~/catkin_ws/src
git clone https://github.com/RobotnikAutomation/zed-ros-wrapper.git
cd zed-ros-wrapper
git submodule update --init --recursive
```

Build the package and its dependencies:

```
cd ~/catkin_ws
rosdep install --from-paths src --ignore-src -r -y
catkin_make
source devel/setup.bash
```

Launch the ZED camera in order to test it: zed.launch, zed2.launch or zed2i.launch

```
roslaunch zed_wrapper zed.launch
```

WARNING: the first time that a new camera is connected, the SDK needs an internet connection in order to get the calibration file. Otherwise, the camera will not run.

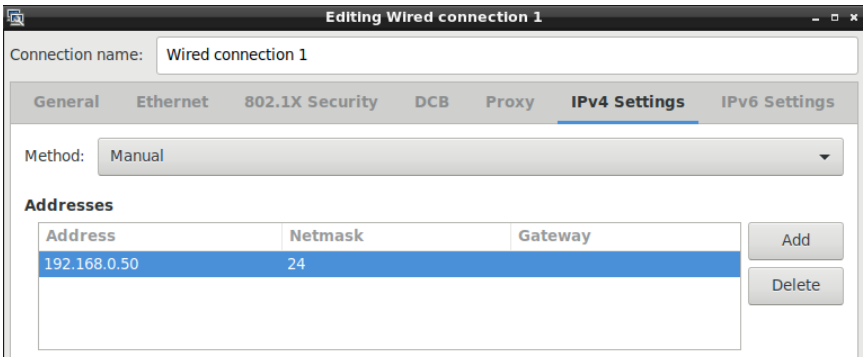
ZED connection -> CALIBRATION FILE NOT AVAILABLE

These are the steps:

1. Connect the camera to the Jetson Nano
2. Connect the Jetson Nano to an internet connection
3. Launch the zed_wrapper package
4. Now the camera can work offline. Repeat the same process for new cameras

7. Jetson Nano Network

Set a static IP. For Jetson Nano must be 192.168.0.50. Then connect the Jetson to the computer of the robot. **Make sure that the communication is Gigabit Ethernet**



Once the IP is set, you can login into the Jetson Nano from the robot PC using SSH:

```
ssh jetson@192.168.0.50
```

8. ZED Automatic Setup

The autorun service for the zed launch file and the NTP server can be installed automatically with these scripts. **This script assumes that the IP of the robot is 192.168.0.200.** You can change the IP by editing the script.

From the Jetson Nano

Go to scripts folder of the zed_bringup package

```
roscd zed_bringup/scripts
```

Run the jetson script

```
./zed-jetson-setup.sh
```

From the robot PC

Go to scripts folder of the zed_bringup package

```
roscd zed_bringup/scripts
```

Run the robot script.

```
./zed-robot-setup.sh
```

The next time the Jetson Nano boots, the bringup script will run. **Make sure that the Jetson Nano is connected to the robot PC, otherwise it will not find the time server**

For bringup.sh of robot_bringup package **one more step is required**. Enable ssh login without password. In this way, the bringup of the robot can run the bringup of the jetson nano.

```
ssh-keygen -t rsa
```

Just press enter when it asks for the file, passphrase, same passphrase.

```
ssh-copy-id jetson@192.168.0.50
```

9. ZED Manual Setup

These are the instructions for the manual setup if the automatic script does not work for your application:

9.1 Jetson Nano steps:

Install screen and nano dependencies:

```
sudo apt-get install screen
sudo apt-get install nano
```

Create a ~/.screenrc file inside /home/jetson directory

```
sudo nano ~/.screenrc
```

Then copy and paste the following configuration:

```
termcapinfo xterm* ti@:te@
shell -$SHELL
setenv LD_LIBRARY_PATH `echo $CMAKE_PREFIX_PATH | awk '{split($1, a, ":"); print
a[1];}'`"/lib":/opt/ros/melodic/lib:/opt/ros/melodic/lib/x86_64-linux-gnu
zombie kr
verbose on
```

Go to ros-zed-wrapper folder:

```
cd catkin_ws/src/ros-zed-wrapper/scripts
```

Copy the bringup script into the /home/jetson directory:

```
cp bringup.sh $HOME
```

Create and enable the autorun service:

```
sudo cp jetson-ros.service /etc/systemd/system
sudo systemctl enable jetson-ros.service
```

Install the NTP server

```
sudo apt-get install ntpdate
```

Edit the **/etc/hosts** file.

```
sudo nano /etc/hosts
```

Add the host name of the NTP server. The IP address must be the static IP of the robot

```
192.168.0.200       robot-time-server
```

Go to ros-zed-wrapper folder:

```
cd catkin_ws/src/ros-zed-wrapper/scripts
```

Add the timesyncd configuration

```
sudo cp ntp/timesyncd.conf /etc/systemd/timesyncd.conf
```

And configure the service:

```
sudo systemctl stop ntp
sudo systemctl disable ntp
sudo systemctl enable systemd-timesyncd
sudo systemctl start systemd-timesyncd
```

Finally, copy the ntp-seconds script into the /home/jetson directory:

```
cp ntp/ntp-seconds.sh ~/ntp-seconds.sh
```

The next time the Jetson Nano boots, the bringup script will run. **Make sure that the Jetson Nano is connected to the robot PC, otherwise it will not find the time server**

Notes: the bringup script creates a screen called zed and establishes connection with the roscore running on the robot PC. By default ROS_MASTER_URI is 192.168.0.200 (robot) whileas ROS_IP is 192.168.0.50 (jetson nano). If your robot or jetson nano has another IP, you will have to edit the bringup script with the correct IPs.

10. Jetson Nano and roscore connection

Jetson Nano runs the ROS package of the ZED camera in the roscore of the robot. When the roscore of the robot is down (for example, when the user runs the bringup script manually), the Jetson Nano loses the connection. In that case, the zed screen of the jetson nano must be launched again.

10.1 Jetson Nano reconnection

Use the SSH connection:

```
ssh jetson@192.168.0.50
```

Secondly, run the bringup script again

```
./bringup
```

Now, the ROS ZED nodes should be visibles again from the robot PC.

10.2 Robot first boot

If the robot PC and the jetson nano boots at the same time (for example when the robot is turned on completely). The connection will work since the jetson nano waits for a roscore before launching the ZED camera. In case that it does not work, you can use the 9.1 Jetson Nano reconnection.

Legacy configuration

Nvidia Jetson Nano Setup

Note: You can find the last packages and image for the jetson here.
si
To power on the device, its DC jack output was used, which required the insertion of a jumper in the board.

The installation steps followed were based on this guide:
https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit

Note: after installation Ubuntu desktop environment was set as Unity did not function well for some reason.

Computer name: jetson-robotnik

Login: jetson
Password: R0b0tn1K

IP was manually set to 192.168.0.50, to ensure proper connectivity to Kairos.

ROS Melodic, ZED SDK and wrapper were installed, according to the instructions found in:
https://www.stereolabs.com/blog/ros-and-nvidia-jetson-nano/

Currently, default ZED parameters (720p, 30Hz) are set. World, odom and base_link frames were set in the zed_wrapper common.yaml configuration file. In addition, the transform between rbkairos_a_base_link and zed_camera_center was specified in the urdf file of the package.

To ensure that the ZED main launch file would be run automatically, an autologin scheme was configured (see corresponding Robotnik guide). The following lines were included in its .bashrc:

```
# ROS
source /opt/ros/melodic/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://kairos-demo:11311

echo "JETSON-ROBOTNIK"
        Terminal=`tty`
```

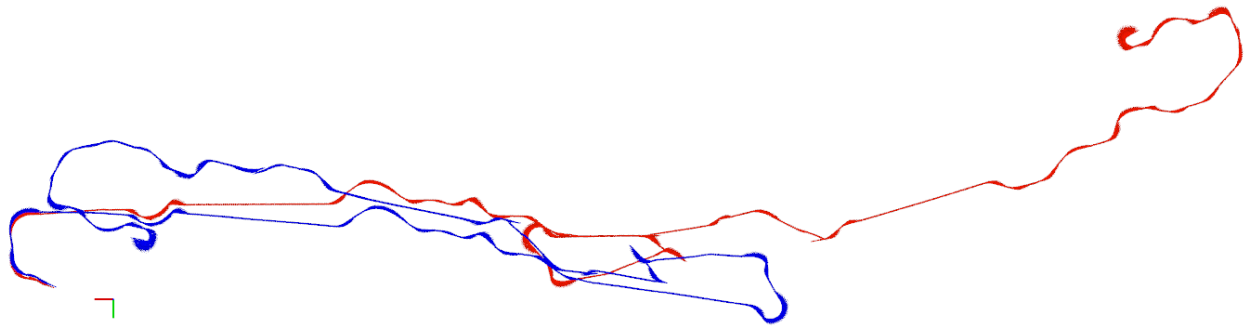
case $Terminal in
"/dev/tty2") sleep 10; roslaunch zed_wrapper zed.launch;;
esac

Due to power limitations (Jetson + ZED occasionally require more than 2A together), Kairos's DC converter was switched to one with a 6A capacity. **Update**: it seems that the problem was with the cable.

Now, images (in compressed mode) can be retrieved at 30Hz, with the exception of depth ones whose compression is demanding. An attempt was made to increase throughput by reducing PNG compression level, however it still barely can reach 7Hz. To be investigated.

ZED Odometry is accessible by echoing the topic /zed/zed_node/odom. 2D mode was set, with spatial alignment option turned on (which means loop closures can occur).

Below is a brief comparison of robot odometry (in red) versus ZED odometry (in blue) for a single lap around Robotnik's garage:



As seen, ZED odometry performs better.

Time Sync

Time synchronization is mandatory to be able to work between the Jetson and the Main PC.

You can read about it here:

- https://vitux.com/how-to-install-ntp-server-and-client-on-ubuntu/
- https://www.digitalocean.com/community/tutorials/how-to-set-up-time-synchronization-on-ubuntu-16-04
- https://feeding.cloud.geek.nz/posts/time-synchronization-with-ntp-and-systemd/

To sum up you have to add the NTP server (The robot pc) in the file /etc/ntp.conf as a ntp server of the jetson.

```
pool SXLSK-190911AA prefer iburst
pool 0.ubuntu.pool.ntp.org iburst
pool 1.ubuntu.pool.ntp.org iburst
pool 2.ubuntu.pool.ntp.org iburst
pool 3.ubuntu.pool.ntp.org iburst
```

Run the following commands:

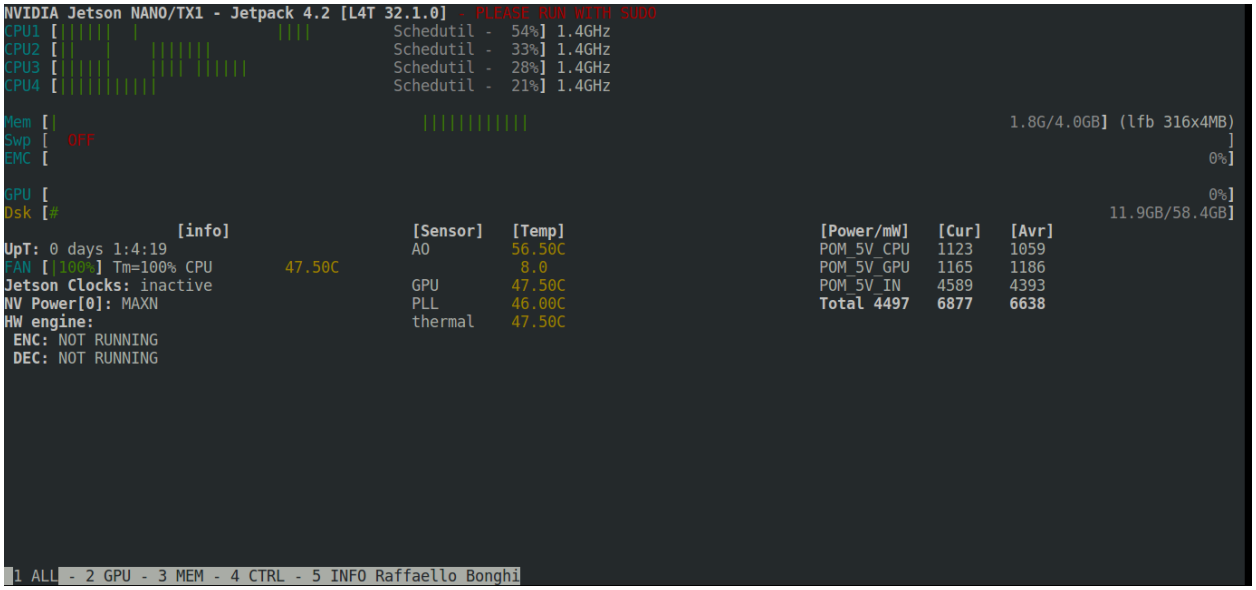
```
> timedatectl set-ntp true
> sudo service ntp restart
> timedatectl
                      Local time: vie 2020-02-28 16:01:18 CET
                  Universal time: vie 2020-02-28 15:01:18 UTC
                        RTC time: vie 2020-02-28 15:01:18
                       Time zone: Europe/Madrid (CET, +0100)
       System clock synchronized: yes
systemd-timesyncd.service active: yes
                 RTC in local TZ: no
```

Monitor GPU stats

More info at https://github.com/rbonghi/jetson_stats

Install:

```
sudo -H pip install jetson-stats
sudo jtop
```



IMPORTANTE

https://answers.ros.org/question/269456/roslaunch-and-ssh-over-local-and-remote-machines-using-avahi/

El launch remoto de la ZED sobre la jetson no funciona correctamente, al menos hasta el momento.

Es importante que la comunicación sea Gigabit Ethernet

https://www.jetsonhacks.com/2020/08/08/clone-sd-card-jetson-nano-and-xavier-nx/