20.02.20

Recherche travaux existents

* [AutoRCCar](https://github.com/hamuchiwa/AutoRCCar)

<https://github.com/hamuchiwa/AutoRCCar>

* [Self-Driving-Autonomous-Car-using-Open-CV-and-Python-Neural-Network-Overtaking-Raspberry-Pi](https://github.com/RobinRajSB/Self-Driving-Autonomous-Car-using-Open-CV-and-Python-Neural-Network-Overtaking-Raspberry-Pi)

<https://github.com/RobinRajSB/Self-Driving-Autonomous-Car-using-Open-CV-and-Python-Neural-Network-Overtaking-Raspberry-Pi>

* Building a self driving RC car by Bert Jan Schrijver and Tim van Eijndhoven:

<https://www.youtube.com/watch?v=OL0vg1WmI6I>

Discussion avec M. Bressy

Faire un planning/commander assez vite la voiture/commencer par le suivit de route

26.02.20

Technologies de détection de l’environnement :

* Caméra
* Lidar
* Plus cher
* Plus gros
* Plus de puissance de calcule
* Plus de courant
* Souvent une seule couche de recherche
* <https://www.dfrobot.com/product-1125.html>
* Très compliqué de détecter les lignes

Technologie de détection d’obstacle :

* Lidar (voir plus haut)
* Laser (VL53L0X)
* 3-120 cm
* Angle <3°
* Optique (Seeedstudio Grove - TF Mini LiDAR)
* 30-1200 cm
* Résolution 1cm
* Angle 2.3°
* Ultrason (HC-SR04)
* 2-400cm
* Résolution 3mm
* Angle de détection 30°
* Peut-être perturbé à l’extérieur

27.02.20

Voiture RC:

* Demander directement au vendeur
* Pouvoir recharger sans changer batterie

Ordinateur de bord

* Temps pour démarrer un RaspBerry, 7-25s

<https://raspberrypi.stackexchange.com/questions/320/what-is-a-typical-boot-time-for-the-standard-debian-distribution-on-a-typical-sd>

* Déterminer s’il faut un microcontrôleur ou microprocesseur

05.03.20

Choix du SBC:

<https://www.hackster.io/news/benchmarking-machine-learning-on-the-new-raspberry-pi-4-model-b-88db9304ce4>

<https://www.hackster.io/news/benchmarking-tensorflow-and-tensorflow-lite-on-the-raspberry-pi-43f51b796796>

<https://www.seeedstudio.com/blog/2019/11/20/best-single-board-computers-of-2019/>

<https://www.seeedstudio.com/blog/2019/12/05/rk3399pro-vs-raspberry-pi-4-vs-jetson-nano-ai-and-deep-learning-capabilities/>

* Jetson Nano Developer Kit
* Coral dev board
* Raspberry Pi 4 + Coral USB Accelerator
* LattePanda Alpha 864s
* ROCK PI N10 Model B

11.03.20

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PWM Freq | Duty cycle | | |
|  | Low | Middle | High |
| Steering | 50Hz | 0.9 (Right) (4.5%) | 1.5 (7.5%) | 2.1 (Left) (10.5%) |
| Speed | 50Hz | 1 (Back) (5%) | 1.5 (7.5%) | 2 (Forward) (10%) |

18.03.20

Connection mannette BT

<https://core-electronics.com.au/tutorials/using-usb-and-bluetooth-controllers-with-python.html>

Liste /dev/input/

* mice : de base, surement si une souris
* event0 : Actif en mode jeu & écran sur le touche présenté ci-dessous

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | gachettes | | | joyL | | | joyR | | | flèches | | | num | | | media | | | select | | | power | | | start | | |
| mice |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| event0 | x |  | x | x |  | x | x |  | x | x |  | x | x |  | x |  |  |  |  |  |  |  |  |  | x |  | x |
| event1 |  |  |  |  |  |  |  |  |  |  | x |  |  |  |  | x | x | x | x | x |  | x | x |  |  |  |  |
| event2 |  |  |  |  | x |  |  |  |  |  | x |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| js0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| mouse0 |  |  |  |  | x |  |  |  |  |  | x |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |

Mise a jours Raspberry 4

<http://wiringpi.com/wiringpi-updated-to-2-52-for-the-raspberry-pi-4b/>

# Instalation raspberry

* Installer raspian sur la carte

<https://www.raspberrypi.org/documentation/installation/installing-images/>

* Connection SSH

<https://raspberry-pi.fr/connecter-ssh-raspberry-pi/>

* Ajouter un fichier nommé « ssh » dans la partition boot pour l’activer
* Insérer la carte dans raspberry
* Installer un client SSH sur le PC (Putty sur Windows)
* Connecter PC à RB via câble Ethernet
* Connecter en SSH avec l’IP « raspberrypi.local » et port 22
* Activer VNC
* Lancer la config du RP

*$ sudo raspi-config*

* Enable VNC dans : 5 Interfacing -> P3 VNC
* Activer le desktop et configurer un écran (vu que aucun écran n’est connecté par HDMI)

3 Boot Options -> B1 Desktop/CLI -> B4 Desktop Auto

7 Advanced Options -> A5 Resolution -> 1920x1080

* Configuration RB via le desktop (pour utiliser le wisard)
* Suivre les instructions de configuration
* Connection auto WiFi (a voir s’il y avait vraiment besoin)

<https://weworkweplay.com/play/automatically-connect-a-raspberry-pi-to-a-wifi-network/>

* Sécurité : Copier le fichier *interfaces*

*$ sudo cp /etc/network/interfaces /etc/network/interfaces\_copy*

* Ouvrir le fichier *interfaces*

*$ sudo nano /etc/network/interfaces*

* Ajouter :

*auto wlan0*

*allow-hotplug wlan0*

*iface wlan0 inet dhcp*

*wpa-conf /etc/wpa\_supplicant/wpa\_supplicant.conf*

*iface default inet dhcp*

* Sécurité : Copier le fichier *wpa\_supplicant.conf*

*$ v /etc/wpa\_supplicant/wpa\_supplicant.conf\_copy*

* Installer Samba

<http://nagashur.com/blog/2016/07/21/partage-de-fichiers-samba-avec-le-raspberry-pi/>

<https://pimylifeup.com/raspberry-pi-samba/>

* Mettre a jour le system

*$ sudo apt-get update && sudo apt-get upgrade && sudo apt-get dist-upgrade*

* Installer Samba

*$ sudo apt-get install samba samba-common-bin*

Oui pour active Winds si demandé

* Configurer Samba
* Sécurité : Faire une copie de smb.conf

*$ sudo cp /etc/samba/smb.conf /etc/samba/smb.conf\_copy*

* Ouvrir le fichier smb.conf

*$ sudo nano /etc/samba/smb.conf*

* Sous [homes], modifier :

browseable = yes

read only = no

create mask = 0777

directory mask = 0777

* Sauver
* Ajouter un User

*$ sudo smbpasswd -a pi*

Puis rentrer 2x le MDP raspberry

* Restart Samba

*$ sudo systemctl restart smbd*

* Ajouter un lecteur réseau Windows avec l’adresse [\\raspberrypi\homes](file://\\raspberrypi\homes)
* Connection Bluetooth de la manette XBOX

<https://pimylifeup.com/xbox-controllers-raspberry-pi/>

<https://core-electronics.com.au/tutorials/using-usb-and-bluetooth-controllers-with-python.html>

# Utilisation de la manette

<https://python-evdev.readthedocs.io/en/latest/index.html>

# Installation de OpenCV

Attention à la version de OpenCV (marche plus depuis 4.1.1) et écrire pip3 au lieu de pip

*pip3 install opencv-contrib-python==4.1.0.25*

<https://www.pyimagesearch.com/2019/09/16/install-opencv-4-on-raspberry-pi-4-and-raspbian-buster/>

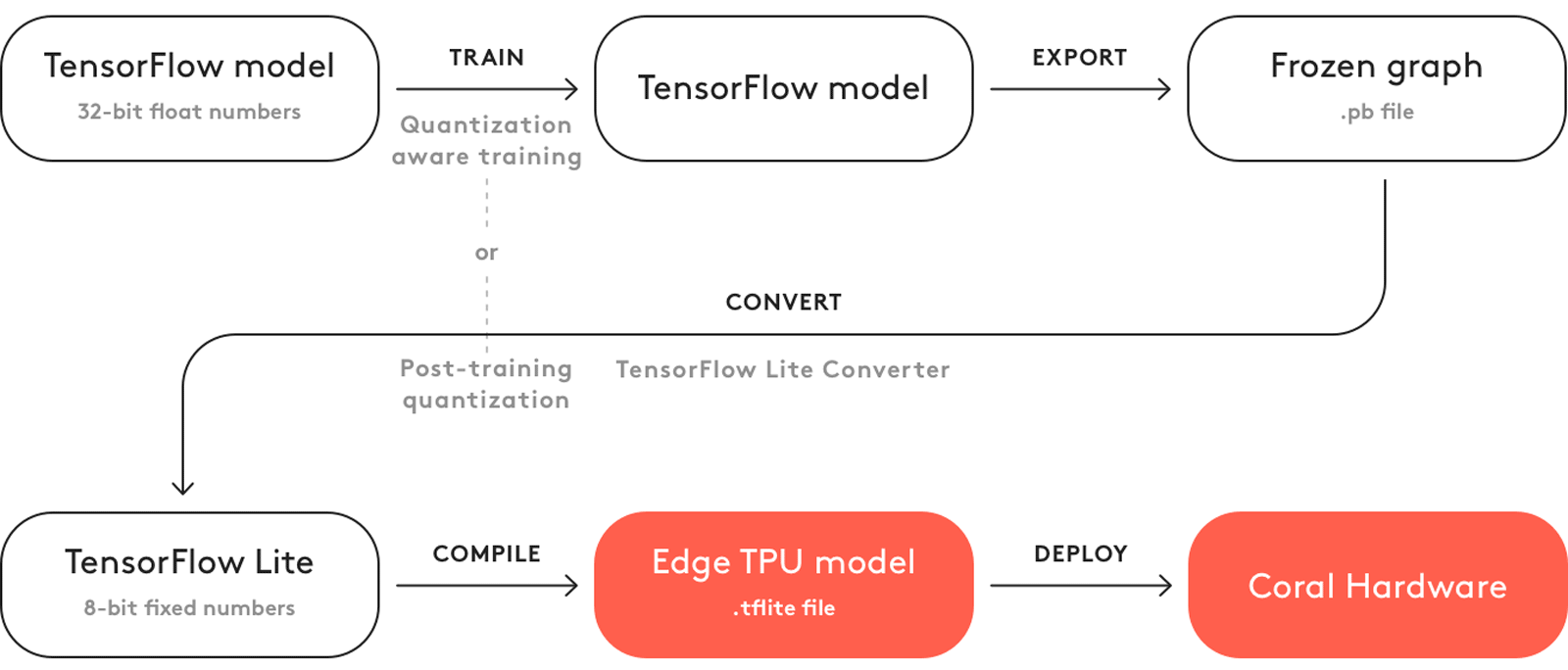
# 30.03.20

## TensorFlow

<https://coral.ai/docs/accelerator/get-started/>

<https://coral.ai/docs/edgetpu/models-intro/>

Train the model on TensorFlow and then convert to TensorFlow Lite



## Test Vitesse USB2 vs USB3

# Installation camera Arducam

<https://www.arducam.com/docs/cameras-for-raspberry-pi/mipi-camera-modules/camera-userland-driver-sdk-and-examples/>

## Pour avoir l’image de preview a l’aide de VNC

Sur le Raspberry -> VNC Connect (en haut à droite) -> Option -> Troubleshooting -> Enable direct capture mode

## Comparaison Arducam vs picam



Figure 1 Arducam

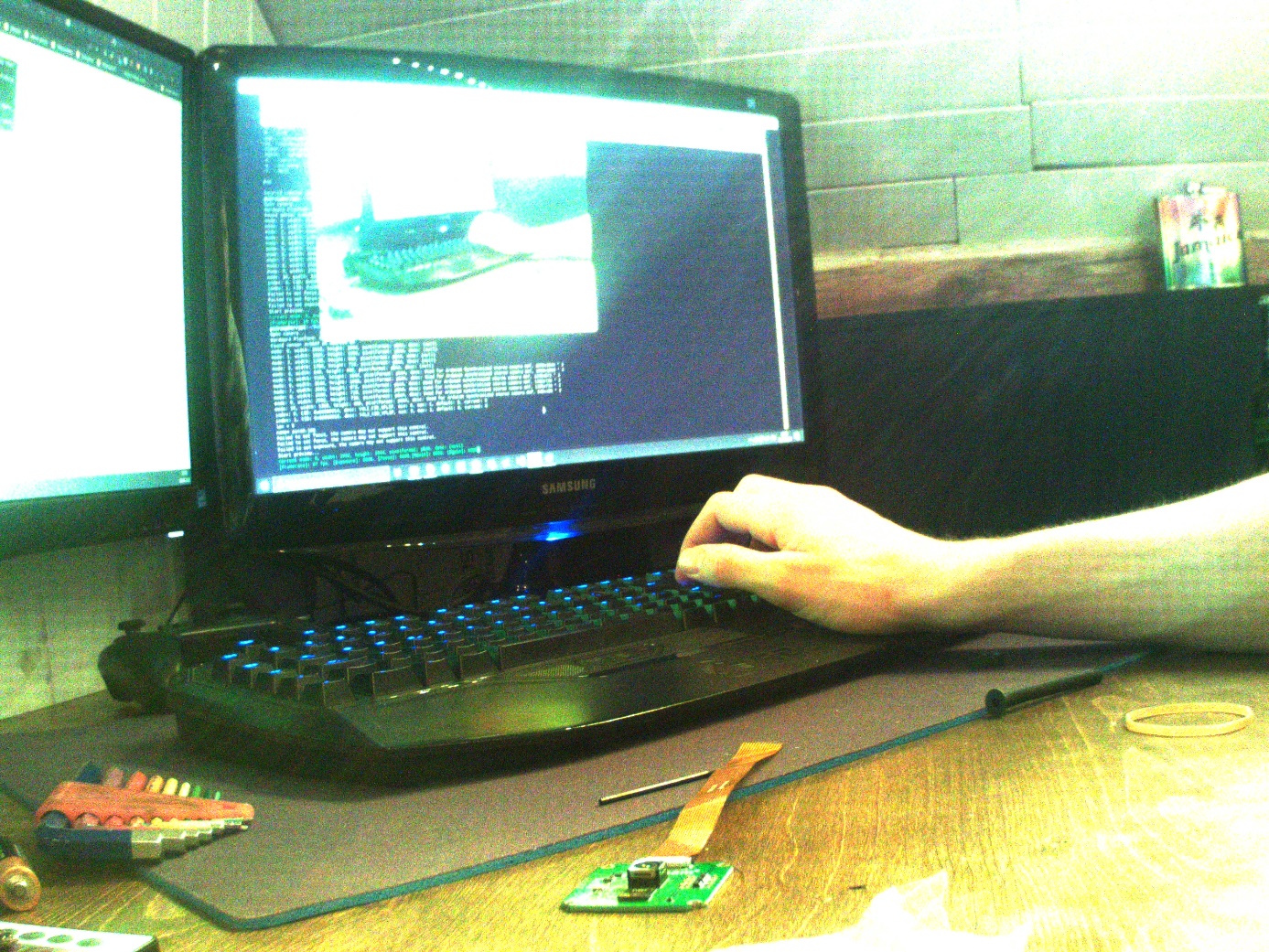


Figure 2 Picam

# Construction de route

Largeur voiture standard : 1.8m

<https://mobilitepietonne.ch/wordpress/wp-content/uploads/2017/07/06_2017_Fiche-info_Cas_de_croisement.pdf>

Largeur chaussée autoroute : 3.5m

<http://piece-jointe-carto.developpement-durable.gouv.fr/REG074B/FONCIER_SOL/N_OCCUPATION_SOL/L_EMPRISE_ROUTE_R74/Fiche1-7-1.pdf>

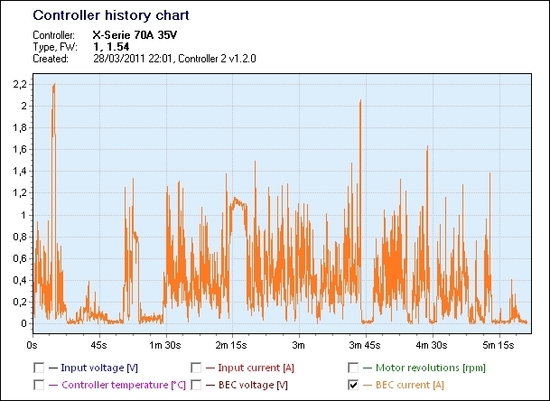
Rapport

Largeur voiture RC : 30cm

Largeur route miniature :

# 09.04.2020 – Montage mécanique + électronique

Consommation courant servo :

[http://rcx-team.kazeo.com/consommation-d-un-servo-a121027458](http://rcx-team.kazeo.com/consommation-d-un-servo-a121027458)

# 11.04.20

## Installation VS Code – Remote SSH

<https://code.visualstudio.com/docs/remote/ssh#_getting-started>

## Utilisation de la lib Picamera

<https://picamera.readthedocs.io/en/release-1.10/index.html>

# 12.04.20

## Electronique

Branchement du servo de steering directement sur la batterie de la voiture et non sur le RPi car sinon trop de courant est tiré et la RPi s’éteint

## Soft

Lors de la capture d’une image qui sera mise dans un tableau numpy, la résolution horizontale doit être un multiple de 32 et la verticale doit être un multiple de 16.

C’est pour ceci que la résolution de la caméra est fixée à (2592, 1920) et non (2592, 1944)

<https://picamera.readthedocs.io/en/release-1.12/recipes2.html#capturing-to-a-numpy-array>

# 14.04.20

Etant donné que la caméra est une caméra IR, l’image est plus violette. Il faudrait mettre un filtre anti IR, mais nous ne pourrions plus avancer dans la nuit.

# 17.04.20

Le RPi à couper quand la tension de la batterie était à 3.5V

Mise en place d’une tache cron exécuté toutes les minutes pour sauvegarder l’état de la batterie afin d’en faire un graphique

chmod u+x /home/pi/Documents/AutonomousRcCar/ARC\_Code/batteryInfoSaver.py

crontab -e

\* \* \* \* \* /home/pi/Documents/AutonomousRcCar/Code/printBatteryInfos.py >> /home/pi/Documents/AutonomousRcCar/BatteryInfo/BatteryInfoError.txt >> /home/pi/Documents/AutonomousRcCar/BatteryInfo/BatteryInfoSave.csv

# 18.04.20

## Construction route

60cm

120cm

## Calibration camera

<https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html>

Prise d’image :

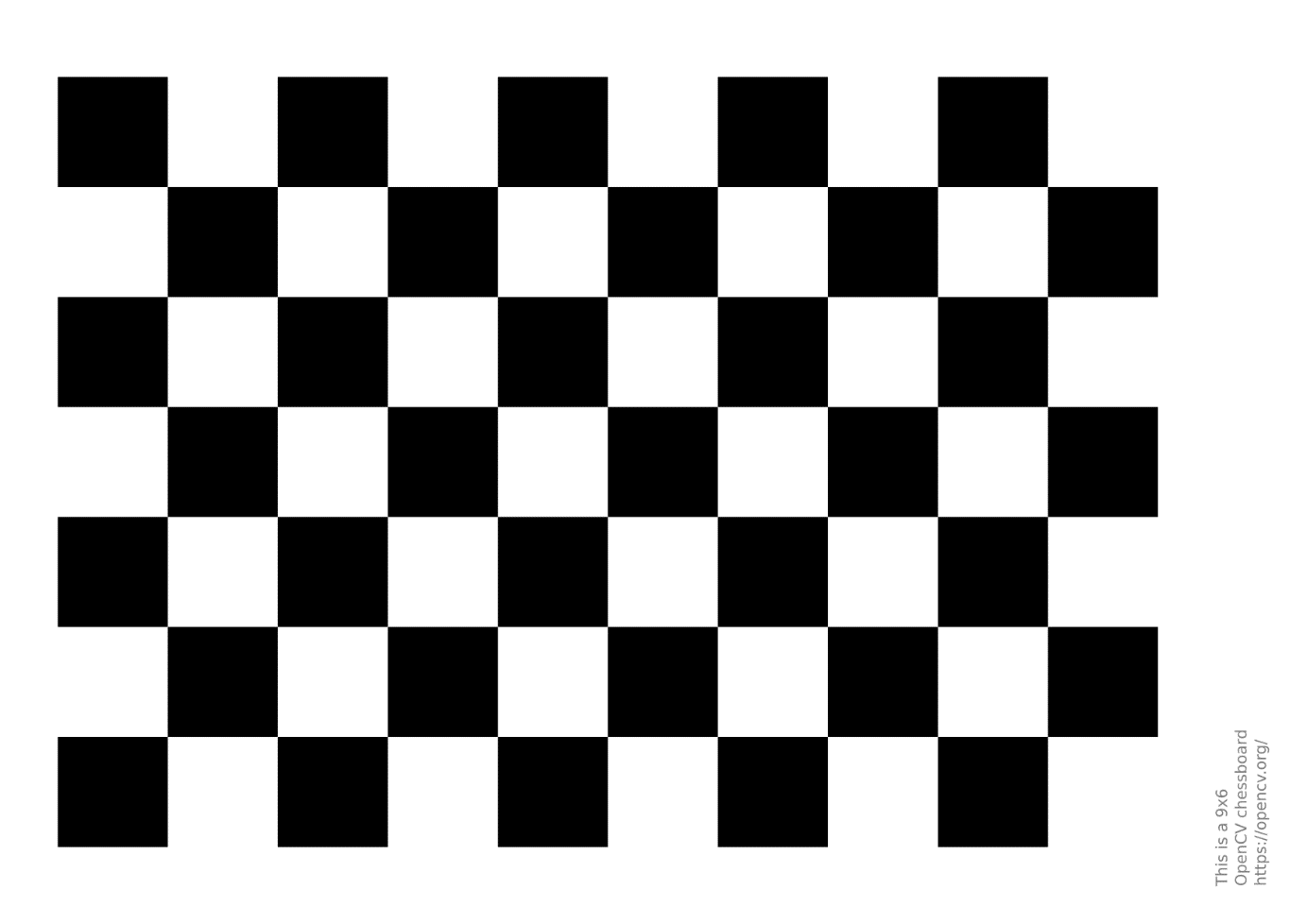
raspistill -o /home/pi/Documents/AutonomousRcCar/ARC\_Code/CameraCalibration/Images/1.jpg -q 100 -e jpg -w 2592 -h 1944

# 23.04.20

Il y a deux semaines, j’avais des problèmes de batterie. Lorsque le servo fonctionnait, le RPi s’arrêtait. Il tirait trop de courant. J’ai monté le servo directement sur la batterie de la voiture.

## Calibration

Longueur 10 carrés : 245mm -> Taille carré 24.5 x 24.5 mm



# 24.04.20

Capture video slowmotion

raspivid -o /home/pi/Videos/1.h264 -t 10000 -w 640 -h 480 -md 6 -fps 60 -pts /home/pi/Videos/timestamps1.txt

Merge le timestamp avec video

mkvmerge -o /home/pi/Videos/bb.mkv --timecodes 0:/home/pi/Videos/timestamps1.txt /home/pi/Videos/1.h264

# 28.04.20

## Calibration

Pas concluant, mais je continue comme ca pour l’instant, pour avancer

## Perspective warp

Coordonnée des 4 points en partant de en haut à gauche et éguille d’une montre :

(1052, 775),(1550, 775),(95, 1876),(2588, 1876)



# 30.04.20

## Calibration

### V1

Camera tournée vers le haut

Orientation et distance à la caméra quelconque

Damier 6x9

### V2

Camera position basse

Damier sur un plan perpendiculaire à l’axe de la caméra

Damier 4x8

### V3

Orientation et distance à la caméra quelconque

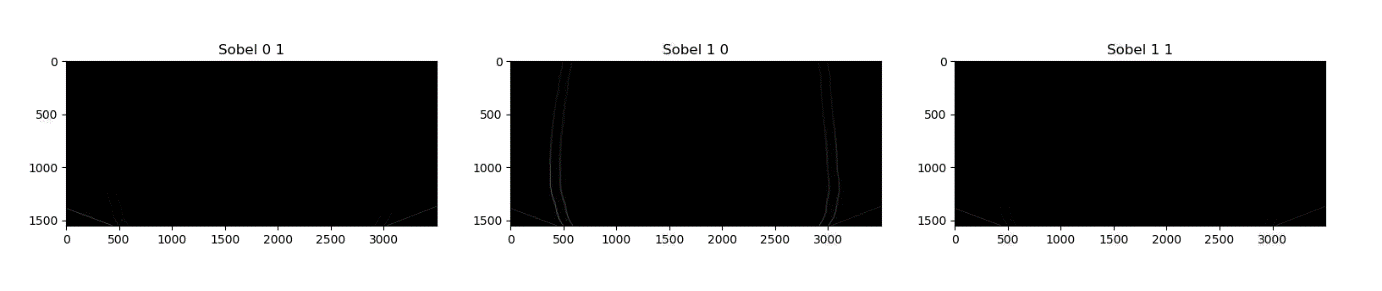
### V2andV3

Combinaison des images des deux versions

# 10.05.20

## HSV HSL différence

## Paramètres du filtre de sobel



# 11.05.20

Afin de pouvoir détecter les pixels appartement à une ou l’autre ligne, nous pouvons utiliser la technique montrer sur le site <https://www.hackster.io/kemfic/curved-lane-detection-34f771> mais dans notre cas de ligne continue il est plus simple de catégoriser les pixels connecté.



# 12.05.20

## Technique d’isolation des lignes :

* Threshold HSV -> Sensible si changement de luminosité / Couleur des lignes
* Sobel sur le canal L -> Laisse plus de bruit sur l’image, des artéfacts reste sur toute l’image. Mais moins sensible aux modifications de luminosité

Figure 3 Sobel filter

## Polyfit

Les axes X et Y ont été inversé dans le polyfit, car les lignes étant majoritairement verticale, il était préférable d’avoir l’axe x à la verticale pour ne pas avoir ce genre de résultat

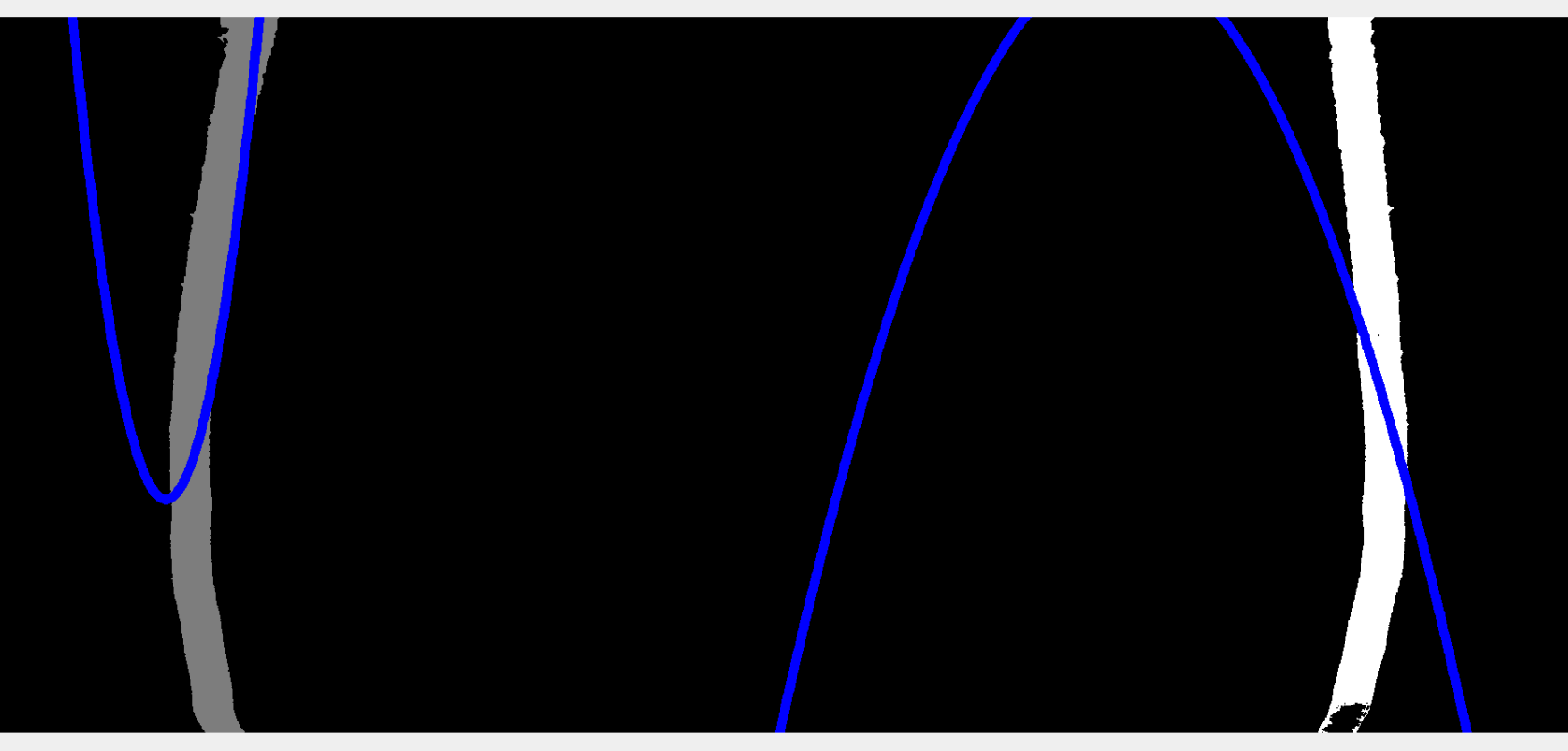


Figure 4 X horizontal



Figure 5 X vertical

# 13.05.20

## Optimisation du temps

### Calibration de l’image

10x | img 2592\*1952 : 6.10s -> 610ms / exec

100x | img 640\*480 : 2.60s -> 26ms / exec

### Warp

100x| img 2592\*1952 : 11.89s -> 119 ms / exec

1000x | img 640\*480 : 7.24s -> 7ms / exec

### BGR to HSV

100x| img 2592\*1952 : 2.71s -> 27 ms / exec

### Threshold

100x| img 2592\*1952 : 3.48s -> 35 ms / exec

### Erode

100x| img 2592\*1952 : 2.35s -> 23 ms / exec

### Connected components

100x| img 2592\*1952 : 7.54s -> 75 ms / exec

### Double Polyfit (right and left)

100x| img 2592\*1952 : 62.53s -> 625 ms / exec

### Full code

100x| img 2592\*1952 : 113.07s -> 1131 ms / exec

1000x | img 640\*480 : 55.58s -> 56ms / exec

## Difference Shape et resolution

Shape: (row, col) donc (y, x)

Resolution : (x, y)