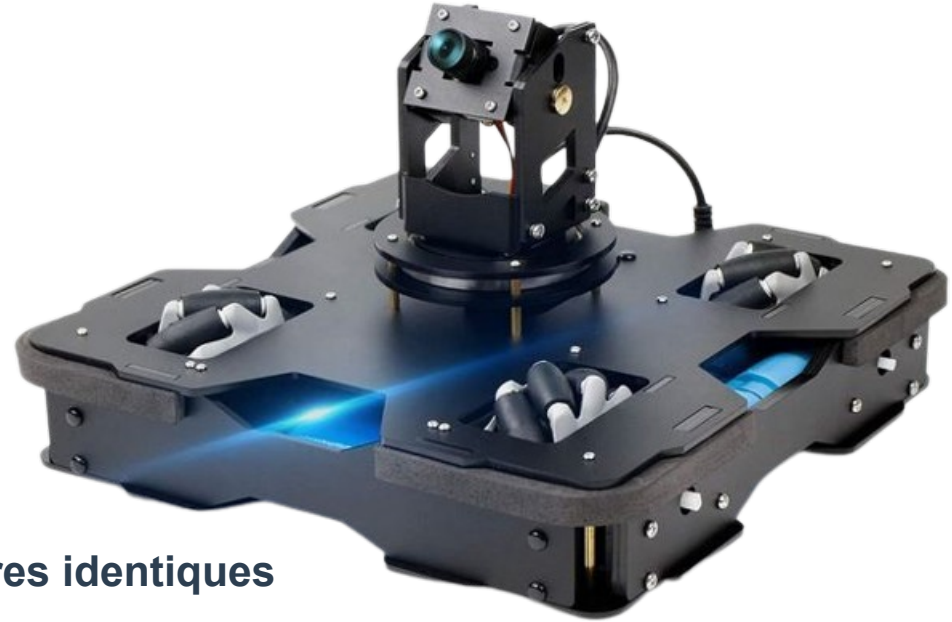
SAE31

Mise en service des Raspblock (16-20/09/2024)

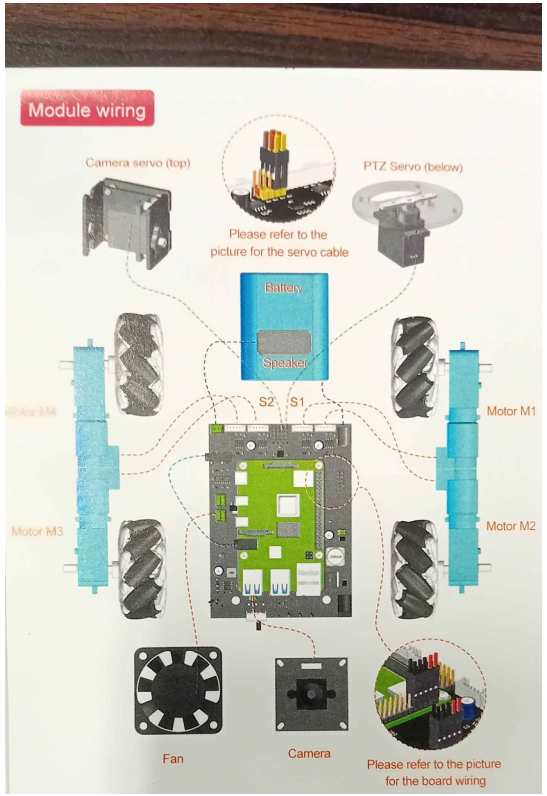
I Objectif

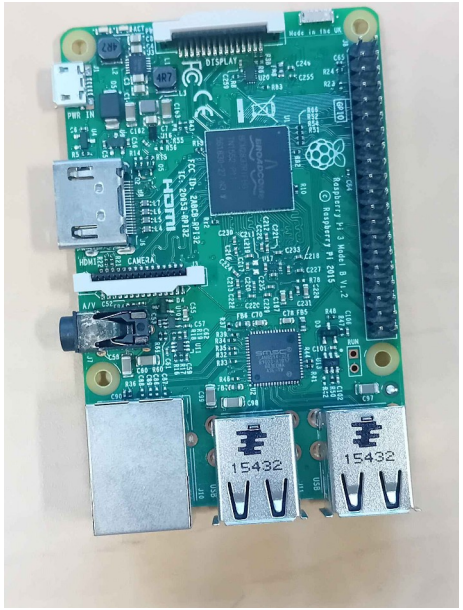
- Se familiariser avec la morphologie des AGV (4-wheeled robots)
- Développer l'environnement de pilotage des raspblock
- Créer un programme de commande de trajectoires identiques à celles des AGV industriels



II Vérification Montage et câblage (jour 1)

RTFM



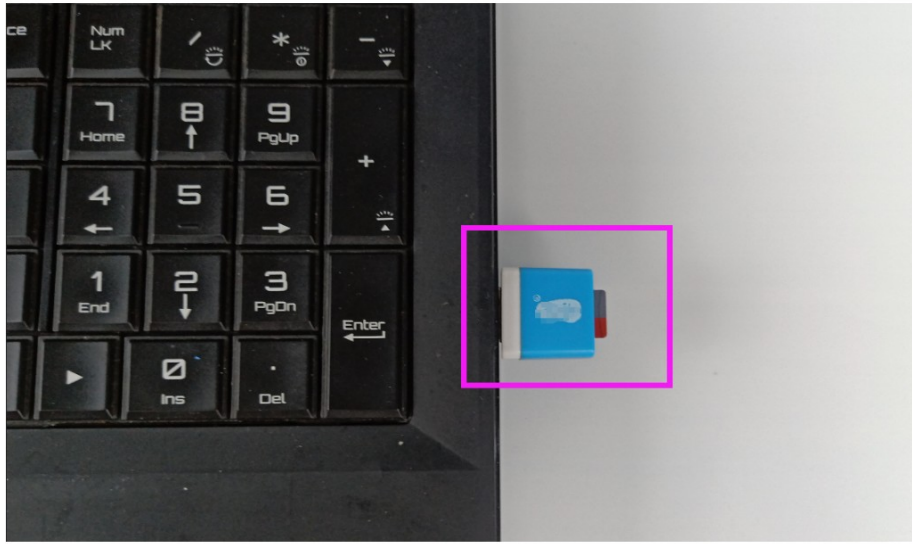


III Mise en place de l'environnement de pilotage du robot (jour 1)

1. Format the SD card:

For windows

Before writing the system image, you need to format your SD card. You need to connect the SD card to the computer with a card reader. As shown below.

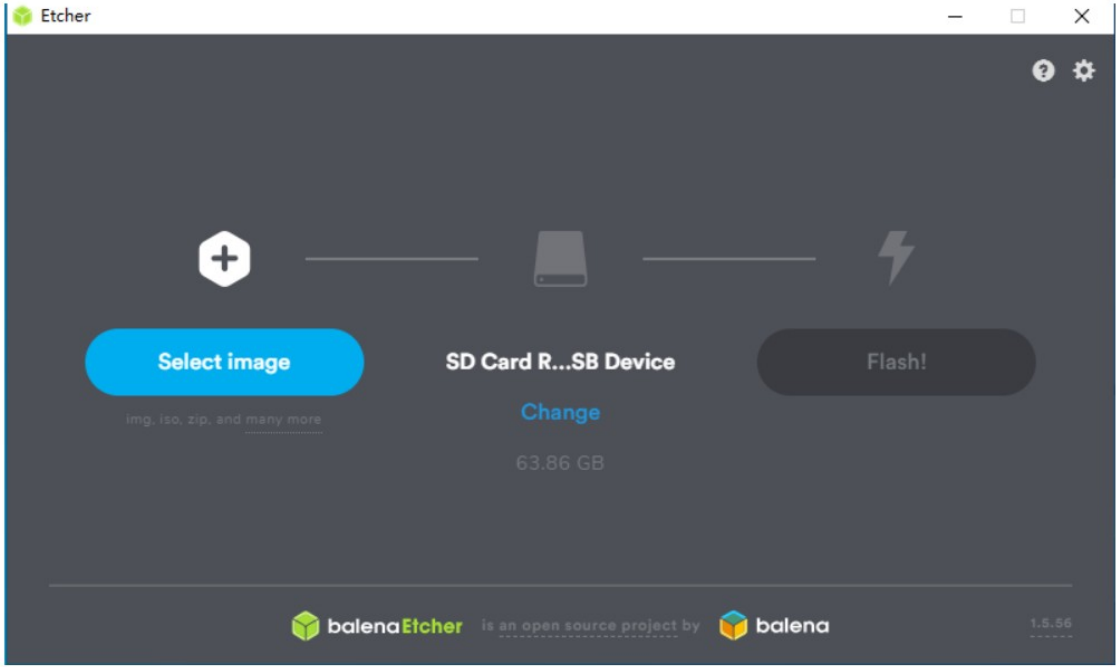


You need to use a tool to format the SD card: **SD Card Formatter 5.0.1 Setup.exe**

2. Write image

For windows

Insert the formatted SD card into the computer through the card reader.



IV apprendre à coder le robot en python (jour 1-3)

Raspblock

1.Installation video
2.Get started
3.Raspberry Pi Basic course
4.Run a single course program
4.1 Preparation
4.2 How to run a single course...
5.OpenCV Basic course
6.Hardware Control cuorse
7. AI Visual course
8. AI Voice course
9.Download APP
10.Battery and charging

Download
Code
Download image

Welcome to Raspblock repository

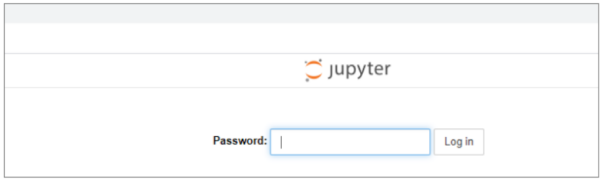
4.2 How to run a single course program

Make Raspblock car and your computer in the same LAN.
1) If our Raspblock car is connected to the same LAN as the PC network via WIFI, we can log in directly on the PC-side Google Chrome or other browser (Raspblock IP address): 8888 to Raspblock's Jupyter Lab,

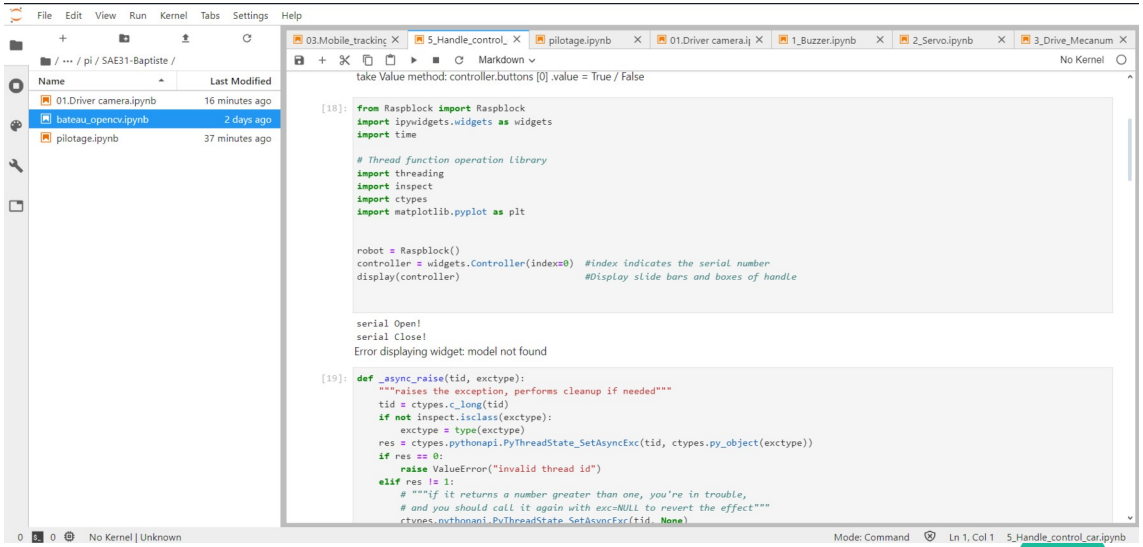
for example: http://192.168.1.244:8888

192.168.1.244 is my Raspblock car IP address. You need to input your own IP address.

As shown below:



2) Input password. If you use image we provided, password is **yahboom**.



IV apprendre à coder le robot en python (jour 1-3)

- **Code de l'exemple qui permet de contrôler la vitesse des 4 roues du robot afin de le manœuvrer dans les directions que l'on veut**

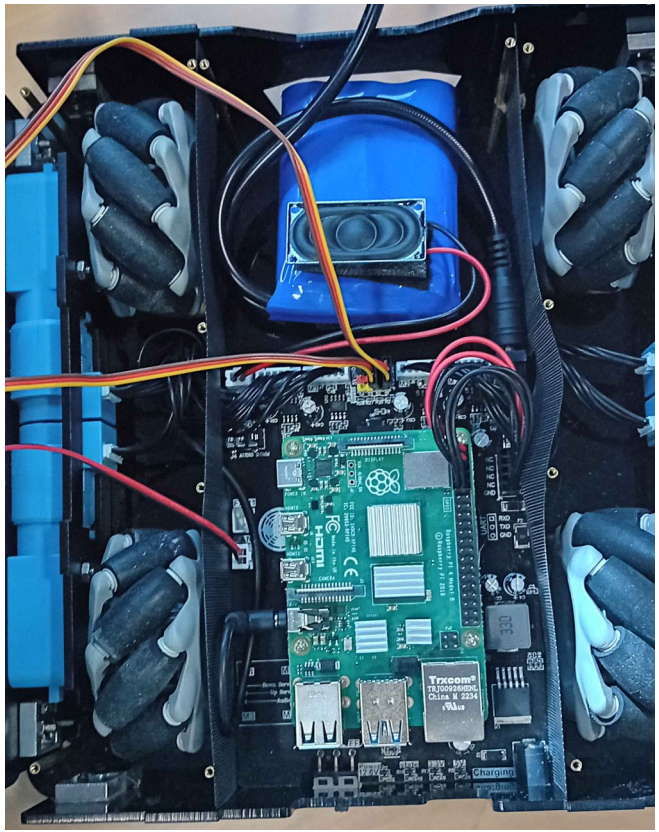
- **Code permettant d'aligner la caméra avec le robot**

```
robot = Raspblock()
robot.Servo_control(2000, 1950)
```

```
[40]: robot.Speed_Wheel_control(4, 0, 0, 0)      #Control wheel individually

[1945]: robot.Speed_Wheel_control(10, 10, 10, 10)      #All wheel forward with 2 speed

[713]: robot.Speed_Wheel_control(-2, -2, -2, -2)  #All wheel reserve with 2 speed

[1180]: robot.Speed_Wheel_control(20, 20, -20, -20)    #Spin left

[953]: robot.Speed_Wheel_control(-1, -1, 1, 1)    #Spin right

[1117]: robot.Speed_Wheel_control(2, -2, 2, -2)    #Left translation

[3004]: robot.Speed_Wheel_control(-10, 1, 1, 10)    #Right translation

[2875]: robot.Speed_Wheel_control(1, -10,10, -1)
        #All wheel stop

[ ]: # Keep moving for a while
     import time
     print ("Start : %s" % time.ctime())
     for i in range(1, 4000):
         robot.Speed_Wheel_control(2, 2, 2, 2)
     print ("End : %s" % time.ctime())
```

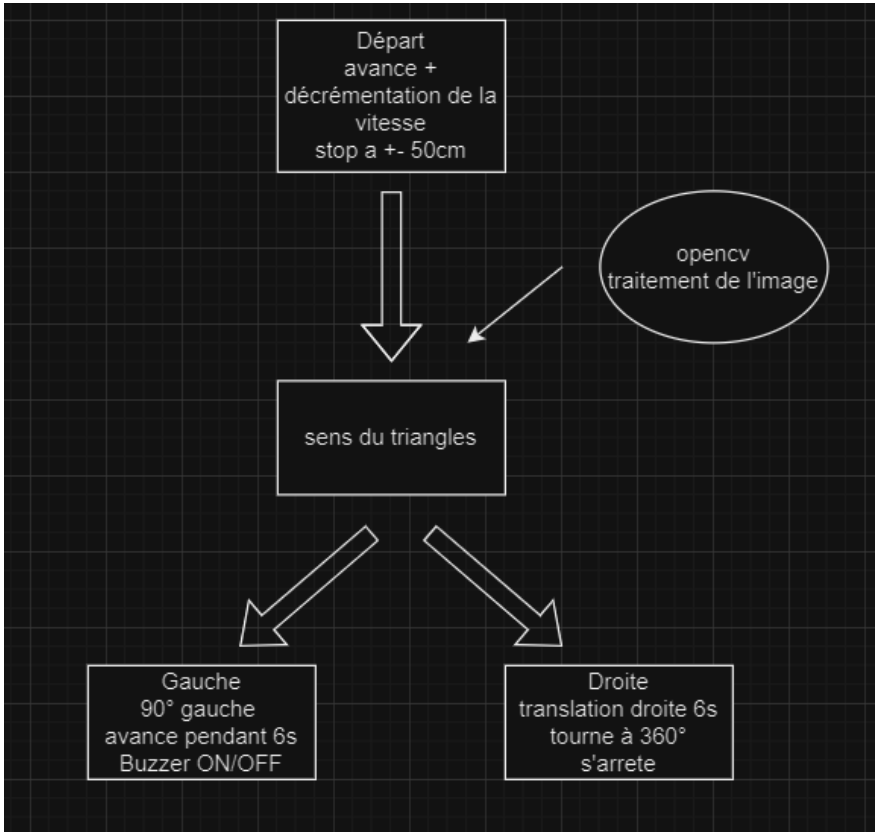
IV apprendre à coder le robot en python (jour 4)

- J’ai changé la Raspberry pi 3 B par une pi 4
  car le ne « tenait » pas les calculs.
- Faut tout débrancher après avoir enlevé la batterie
- Changer la carte microSD de la pi 3 à pi 4
- Remonter la pi 4 dans le Raspblock



IV défi

- **défie suivre se diagramme le mieux possible**



IV défi

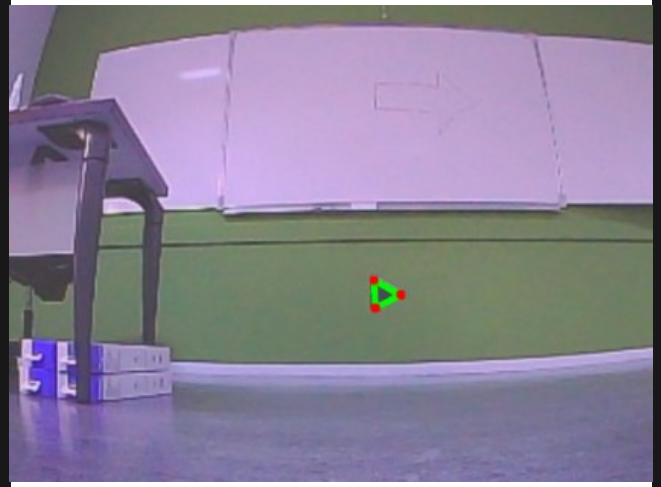
```
def FG():
    robot = Raspblock()
    # On tourne de 90°
    look_forward=2000
    turn_left90 = time.time()
    while time.time()-turn_left90 < 0.66:
        look_forward -= 3
        robot.Servo_control_single(1, look_forward)
        robot.Speed_Wheel_control(8, 8, -8, -8)  #All wheel reserve with 2 speed
    forward10=time.time()
    while time.time() - forward10 <10:
        robot.Speed_Wheel_control(1,1,1,1)
    for i in range(2):
        robot.Buzzer_control(1)
        time.sleep(1)
        robot.Buzzer_control(0)

def FD():
    robot = Raspblock()
    dtranslation=time.time()
    while time.time() - dtranslation <6:
        robot.Speed_Wheel_control(-5,5,-5,5)
    troisix0=time.time()
    while time.time() - troisix0 <2.64:
        robot.Speed_Wheel_control(-8,-8,8,8)
```

Programme qui permet de faire les taches dans les cas de la flèche a gauche et à droites

IV défi

```python
def get_triangle(img):
    # Convertir l'image en niveaux de gris
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Égaliser l'histogramme pour compenser les variations de lumière
    #gray = cv2.equalizeHist(gray)

    # Appliquer un filtre Gaussien pour réduire le bruit
    gray = cv2.GaussianBlur(gray, (3, 3), 0)

    # Appliquer un seuillage adaptatif (moins sensible à l'éclairage)
    thresh = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
                                   cv2.THRESH_BINARY_INV, 11, 2)

    # Utiliser Canny pour détecter les contours
    edges = cv2.Canny(thresh, 50, 150)

    # Trouver les contours dans l'image
    contours, _ = cv2.findContours(edges, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    triangle = [cv2.contourArea(cnt)for cnt in contours]
    # Boucle sur chaque contour détecté
    for contour in contours:
        # Filtrer les petits contours (éliminer le bruit)
        area = cv2.contourArea(contour)
        if area < 200:  # Ajuste ce seuil en fonction de la taille des formes dans ton image
            continue

        # Approximer les contours pour simplifier la forme
        epsilon = 0.04 * cv2.arcLength(contour, True)
        approx = cv2.approxPolyDP(contour, epsilon, True)
        # Si le contour approximé a 3 sommets, c'est un triangle
        if len(approx) == 3:
            # Dssiner le contour du triangle
            cv2.drawContours(img, [approx], 0, (0, 255, 0), 5)
            for point in approx:
                cv2.circle(img, tuple(point[0]), 5, (255, 0, 0), -1)
                return approx
```

Programme qui permet de détecter les triangles ainsi que dire leurs sens (gauche, droite)



```python
def LOR(points):
    listx = []
    for coor in points:
        listx.append(coor[0][0])
    meanx = np.mean(listx)
    L = 0
    print(listx)
    for pointx in listx:
        if pointx < meanx:
            L +=1
        else:
            pass
    if L ==2:
        print("Droite")
    elif L==1:
        print("gauche")
    else:
        print("erreur")
```