

S5i Projet

Labo auto

ATTENTION fragile !

- Ne pas tester votre code sur une table !
- L'auto tombe = Elle brise !

Avant de commencer

- Branchez clavier/souris/écran si requis
- Vérifier l'indicateur de batteries (1 ou 2 lumières vertes!)
- Chargeur et info batteries (voir message des techniciens)
- En premier :
 - Connectez-vous à une borne WiFi

Mise en arrêt

Faire un « shutdown »

avant de mettre l'auto à OFF!

Login

- Utilisateur: pi
- Mot de passe : raspberry

Remote login

- Pour obtenir l'adresse IP
(faire sur le RaspberryPi en commande):
ifconfig
- SSH
 - ssh pi@192.168.0.xxx (adresse de ifconfig)
- VNC
 - installer realVNC (<https://www.realvnc.com>)

Disponibilité des librairies et exemples

- Exemples dans :
~/Documents/SunFounder_PiCar-S
Suggestion : Copiez le répertoire d'exemples
- Autres librairies:
~/Documents/SunFounder_PiCar

Calibration

- Fichier de configuration : config
 turning_offset = 20
 forward_A = 0
 forward_B = 0
- Copier le fichier dans le répertoire de votre application

Exécution

- Utilisez python3

Programmation

- Classes et méthodes
 - front_wheels
 - turn()
 - droit(straight) 90
 - Plage 45 à 135 (+/-45)
 - turn_straigth()

Programmation

– back_wheels

- forward()
- backward()
- speed()
 - 0 à 100
- stop()

Programmation

- Line_Follower

- read_analog()
- read_digital()

- Ultrasonic_Avoidance

- distance()
- get_distance()
 - Paramètre, nombre de lectures
- less_than()
 - Fait 5 mesures avant de comparer

Programmation

- Capteur distance
 - Défaut, moyenne sur 5 mesures
- Capteur de ligne
 - Calibration ligne noire (pour établir le vecteur des références):
 - Démarrer le logiciel de test :
`python3 test_line_module.py`
 - Placer le véhicule au-dessus d'une section pâle
Lire les valeurs des 5 capteurs
 - Placer le véhicule au-dessus d'une section foncée (ligne)
Lire les valeurs des 5 capteurs
 - déterminer la référence entre les 2 séries de valeurs
 - Possible d'ajuster le potentiomètre pour améliorer l'écart

Programmation

- Exemples
 - Juste déplacer

Programmation

```
1  #!/usr/bin/env python
2
3  from picar import front_wheels
4  from picar import back_wheels
5  import time
6  import picar
7
8  picar.setup()
9
10 fw = front_wheels.Front_Wheels(db='config')
11 bw = back_wheels.Back_Wheels(db='config')
12
13 def start_bouge():
14     print('Demarrage')
15
16     fw.turn_straight() # Correspond a 90
17     bw.forward()
18     bw.speed = 70
19     time.sleep(5)
20
21     fw.turn(0)
22
23     bw.backward()
24     bw.speed = 20
25     time.sleep(5)
26
27     fw.turn(180)
28
```

```
29     bw.forward()
30     time.sleep(5)
31
32     fw.turn(90) # tout droit
33
34     bw.forward()
35     bw.speed = 70
36     time.sleep(5)
37
38     bw.stop()
39
40
41 def stop():
42     bw.stop()
43     fw.turn_straight()
44
45 if __name__ == '__main__':
46     try:
47         start_bouge()
48     except KeyboardInterrupt:
49         stop()
50
```

Exemple capteur ultrasonique

```
1 from SunFounder_Ultrasonic_Avoidance import Ultrasonic_Avoidance
2 import time
3
4 UA = Ultrasonic_Avoidance.Ultrasonic_Avoidance(20)
5 threshold = 10
6
7 def main():
8     distance = UA.get_distance()
9     status = UA.less_than(threshold)
10    if distance != -1:
11        print('distance', distance, 'cm')
12        time.sleep(0.2)
13    else:
14        print(False)
15    if status == 1:
16        print("Less than %d" % threshold)
17    elif status == 0:
18        print("Over %d" % threshold)
19    else:
20        print("Read distance error.")
21
22 if __name__ == '__main__':
23     while True:
24         main()
25
```

test_ultrasonic_module.py

Exemple capteur de ligne

```
1 from SunFounder_Line_Follower import Line_Follower
2 import time
3
4 lf = Line_Follower.Line_Follower()
5 while True:
6     ▶ print(lf.read_analog())
7     ▶ print(lf.read_digital())
8     ▶ print(' ')
9     ▶ time.sleep(0.05)
10
11
```

test_line_module.py