

Utilisation Raspberry

Commandes utiles

Commandes de base

- `*id*` : `ros`
- `*password*` : `eurobot2019`
- changer l'heure : `sudo date --set "11 FEB 2020 15:53:00"`
- rentrer dans un fichier : `cd [nom du fichier]`
- revenir en arrière : `cd ..`
- lister les dossiers actuel : `ls`
- ouvrir et écrire dans un script : `nano [nom du script]`
- Arrêter un script sur une console : `ctrl+c`
- Mettre en pause un script : `ctrl+z`
- Relancer le script en arrière plan (très utile) : `bg` après le `ctrl+z`
- Compléter le nom d'un script ou d'un répertoire : `tab`
- Lancer plusieurs commande en meme temps (utile) : `commande1 & commande2`

Commandes ROS

- `*connexion vers arduino*` : `roslaunch rosserial_python serial_node.py /dev/ttyACM1`
- `*publication sur un topic*` : `rostopic pub toggle_led std_msgs/Empty --once`
- `*écoute sur un topic*` : `rostopic echo /topic_name`
- `*lancement du rosbridge*` : `roslaunch rosbridge_server rosbridge_websocket.launch`
- `*lancement du node-red*` : `node-red start`
- Emplacement des scripts : `catkin_ws/src/subMarine/src`
- `*lancer un script depuis n'importe ou*` : `roslaunch subMarine [nom du script]`

Scripts et utilisation

- `command.py`: communique les valeurs de déplacement de Node-red vers arduino (topic ROS: `/command`, topic MQTT: `/command`)
- `servo.py`: commande les servomoteurs de la caméra (topic MQTT: `/servo`)
- `submarine.py`: communique les valeurs de profondeur et de pression vers node red (topic ROS: `/chatter`, topic MQTT: `/chatter`)
- `rpi_camera_surveillance_system.py` : lance le stream de la caméra
- `acceleration.py` : communique les valeurs de l'accéléromètre
- `ip.py` : récupère l'ip de la rasp

Comment lancer le robot

1. Lancer la commande de connexion à l'arduino (ttyACM0 ou ttyACM1 ou ttyACM2 en fonction du port de connexion)

2. Lancer le node-red
3. Lancer le launch file
4. Accéder au node-red en allant sur le navigateur et en tapant comme url : "localhost:1880 " et "localhost:1880/ui" pour le dashboard

Ajouter la détection d'ip automatique

1. Placer le script ip.py dans le répertoire du ros (catkin_ws/src/subMarine/src)
2. Écrire « import ip» au début de tous les scripts
3. Remplacer l'ip dans le main par ip.get_ip()