

Matos

1. Routeur Wifi Cisco RV180W qui g re un r seau BBBRouter :
IP Routeur 192.168.1.11
Login cisco
Pwd Cisco2Cisco
Adressage le PC-Turtlebot est adress  par MAC (20:10:7a:fb:2b:ec → 192.168.1.66)
Adressage le PC-Contr leur est adress  par MAC (b0:5a:da:3d:6c:af → 192.168.1.88)
2. PC-Turtlebot
System Ubuntu 14.04 LTS, 64 bits "Turtlebot-HP"
Login turtlebothp
Pwd turtlebothp
— le PC pour le turtlebot est reconnaissable par les deux pions en dessous qui permettent la fixation sur la partie haute du robot.
3. PC contr leur
System Ubuntu 14.04 LTS, 64 bits
— Pour l'instant PC C.A.
4. Turtlebot (Kobuki base & Kinect)
5. Manette XBox One
6. Base de charge Turtlebot
7. Chargeur PC-Turtlebot
8. Multiprise

1 Mise en fonctionnement

1.1 Brancher le routeur

1.2 D marrage du turlebot

- Placer le PC sur le turlebot
- Allumer le PC et le refermer
- Brancher la base Kobuki
- Brancher la Kinect
- Allumer la base Kobuki

1.3 D marrage du PC-Turtlebot

- Acc der au PC-Turtlebot par NoMachine.
- Ouvrir un terminal

```
username@PC-Turtlebot:~$ printenv | grep ROS
ROS_DISTRO=indigo
ROS_PACKAGE_PATH=/opt/ros/indigo/share:/opt/ros/indigo/stacks
ROS_ETC_DIR=/opt/ros/indigo/etc/ros
```

```

ROS_ROOT=/opt/ros/indigo/share/ros
ROS_MASTER_URI=http://192.168.1.66:11311
ROSLISP_PACKAGE_DIRECTORIES=
ROS_HOSTNAME=192.168.1.66

```

Si une des variables ROS_ n'est pas bien configur , faire les exports qui vont bien :

```

# ~/.zshrc
[...]
source /opt/ros/indigo/setup.zsh
export TURTLEBOT_3D_SENSOR=kinect
export ROS_MASTER_URI=http://192.168.1.66:11311 #Adress where
ROSCORE is running => turtlebot PC
export ROS_HOSTNAME=192.168.1.66 # IP of LOCALHOST
[...]

```

Lancer le soft de gestion du robot :

```

username@PC-Turtlebot:~$ roslaunch turtlebot_bringup minimal.
launch

```

1.4 D marrage du PC controleur

- Allumer le PC.
- Brancher le PC au routeur en filaire
- V rifier la connectivit  par NoMachine

```

username@PC-Control:~$ printenv | grep ROS
ROS_DISTRO=indigo
ROS_PACKAGE_PATH=/opt/ros/indigo/share:/opt/ros/indigo/stacks
ROS_ETC_DIR=/opt/ros/indigo/etc/ros
ROS_ROOT=/opt/ros/indigo/share/ros
ROS_MASTER_URI=http://192.168.1.66:11311
ROSLISP_PACKAGE_DIRECTORIES=
ROS_HOSTNAME=192.168.1.88

```

Si une des variables ROS_ n'est pas bien configur , faire les exports qui vont bien :

```

# ~/.zshrc
[...]
source /opt/ros/indigo/setup.zsh
export TURTLEBOT_3D_SENSOR=kinect
export ROS_MASTER_URI=http://192.168.1.66:11311 #Adress where
ROSCORE is running => turtlebot PC
export ROS_HOSTNAME=192.168.1.88 # IP of LOCALHOST
[...]

```

Lancer le soft de monitoring du robot :

```

username@PC-Control:~$ rqt -s kobuki_dashboard

```

2 Pr paration de la d mo AMCL

Il faut tout d'abord cr er la carte de l'endroit ou l'on afit la d mo. Le plus simple est d'avoir une ar ne dans laquelle personne ne p n tre. La partie 1 doit  tre OK.

2.1 PC-Turtlebot

- Acc der au PC-Turtlebot par NoMachine.
- Ouvrir un terminal

```
username@PC-Turtlebot:~$ roslaunch turtlebot_navigation  
gmapping_demo.launch
```

2.2 PC-Control

- Brancher la manette de xbox
- Ouvrir un terminal

```
username@PC-Control:~$ cd Developpement/turtlebot  
username@PC-Control:~/Developpements/turtlebot$ roslaunch  
demoJPO_PC_Nav_Brest.launch
```

Une interface graphique se lance et permet de visualiser la carte que le robot va construire lorsqu'il se d placera.

Pour d placer le robot, on utilise la manette :

- Appui sur LB pour activation de la t l op ration
- Utilisation du stick en haut a gauche de la manette pour le d placement.
- Si on lache LB, on envoi plus de commande (implique qu'un appui sur LB bloque les commandes envoy es en mode auto)

Lorsque la carte construite est convenable(i.e. pas de grossi re erreur lors de multiples rotations). On va pouvoir la sauvegarder sur le robot pour une utilisation ult rieure. Si ce n'est pas le cas, reprenez au d but de cette partie.

2.3 PC-Turtlebot - sauvegarde de la carte

- Acc der au PC-Turtlebot par NoMachine.
- Ouvrir un terminal

```
username@PC-Turtlebot:~$ rosrunc map_server map_saver -f ./  
nomDeLaCarte
```

La carte est sauvegard e par deux fichiers : nomDeLaCarte.pgm et nomDeLaCarte.yaml. On peut alors  teindre les commandes lanc es sur les 2 PC   la partie 2. On peut aussi tout  teindre...

3 D mo - localisation sur une carte par AMCL

La partie 1 doit  tre faite en premier. La partie 2    t  faite aussi mais les commandes qu'elle implique sont arr t es.

AMCL = Adaptive Monte Carlo Localization, m thode probabiliste de localisation, c'est un filtre   particule qui estime la position du robot   partir d'une carte connue. Pour faire simple, on dit au d part que le robot peut se trouver n'importe ou sur la carte, on disperse toutes les particules sur la carte (imaginez lancer des grains de sables sur la carte, chaque grain est une particule et repr sente une position probable pour le robot). Comme le robot mesure les distances aux obstacles les plus proches (par l'infra-rouge de la kinect), on va pouvoir dire que certains grains de sables ne sont pas des positions possibles : si le robot mesure une distance de 2m devant lui, toutes les particules positionn es   moins de 2m d'un mur peuvent  tre ramen es   cette distance. L'algorithme va, au fur et   mesure des distances que le robot va acqu rir, concentrer les particules vers l'endroit ou le robot se trouve r ellement (bon c'est bien plus compliqu  que  a mais pour JPO ca ira...)

3.1 PC-Turtlebot

On doit tout d'abord sp cifier la carte qui sera utilis e par la d mo. Pour des probl mes de droits utilisateurs, le plus simple est de copier cette carte vers /tmp.

- Acc der au PC-Turtlebot par NoMachine.
- Ouvrir un terminal

```
username@PC-Turtlebot:~$ cp nomDeLaCarte.* /tmp
username@PC-Turtlebot:~$ export TURTLEBOT_MAP_FILE=/tmp/
    nomDeLaCarte.yaml
username@PC-Turtlebot:~$ roslaunch turtlebot_navigation amcl_demo
    .launch
```

3.2 PC-Control

Le fichier   lancer et les instructions pour d placer le robot sont les m me qu'  la partie 2.2 :

- Brancher la manette de xbox
- Ouvrir un terminal

```
username@PC-Control:~$ cd Developpement/turtlebot
username@PC-Control:~/Developpements/turtlebot$ roslaunch
    demoJPO_PC_Nav_Brest.launch
```

Une interface graphique se lance et permet de visualiser la carte que le robot va construire lorsqu'il se d placera.

Pour d placer le robot, on utilise la manette :

- Appui sur LB pour activation de la t l op ration
- Utilisation du stick en haut   gauche de la manette pour le d placement.
- Si on l che LB, on envoi plus de commande (implique qu'un appui sur LB bloque les commandes envoy es en mode auto)

3.3 Utilisation

Tout se fait   partir du PC-Control : Au lancement, il faut donner une premi re estimation de sa position   robot :

- On click sur "2D Pose Estimate"
- On click sur la carte   l'endroit approximatif ou se trouve le robot physiquement, attention, il faut maintenir le click et bouger autour du point pour que la direction de la fl che soit orient e de la m me mani re que le robot.

Ensuite, pour naviguer :

- On click sur "2D Nav Goal"
- On click sur la carte   l'endroit que l'on veut atteindre, le robot va y aller en  vitant les obstacles sur son chemin (peut prendre un peu de temps.)

Il est toujours possible de reprendre le contr le par la manette en appuyant sur LB.

Pour annuler une commande "2D Nav Goal", il faut en envoyer une autre   l'endroit ou se trouve le robot.

3.4 Synth se

1. Le PC-Turtlebot doit avoir deux commandes lanc es :

```
username@PC-Turtlebot:~$ roslaunch turtlebot_navigation
    amcl_demo.launch
```

, et

```
username@PC-Turtlebot:~$ roslaunch turtlebot_bringup  
minimal.launch
```

2. Le PC Control doit avoir deux commande lanc es :

```
username@PC-Control:~$ rqt -s kobuki_dashboard
```

, et

```
username@PC-Control:~/Developpements/turtlebot$  
roslaunch demoJPO_PC_Nav_Brest.launch
```

3. Le PC Control doit avoir deux interfaces graphiques :
— le dashboard qui informe sur l tat du robot
— l'interface de navigation