

## Babyface

La carte d'acquisition audio fonctionne avec les pilotes legacy sous Linux. Pour ce faire, il faut mettre à jour le pilote de la carte à une version supérieure à 200.

Si la carte n'est pas automatiquement reconnue au branchement sous Linux, il faut la passer en mode legacy. Débrancher la carte entièrement et appuyez sur les touches Select et Recall en même temps puis rebrancher la carte. Le pilote Class Compliant est alors démarré sur la carte.

Sur le robot, la carte est reconnue en tant que hardware numéro 2 (1 pci intégré+hdmi, 2 babyface, 3 webcam si branchée).

Utiliser Audacity ou arecord (cf. man arecord) pour tester la carte.

Les paquets ros-indigo-audio-common sont installés sur le robot.

```
sudo usermod -a -G audio user
```

## Webcam

alias kCamera

Sur le pc kemar, l'utilisateur kemar a été ajouté à la liste des utilisateurs autorisés à faire de la capture vidéo (sudo usermod -a -G video user).

```
roslaunch usb_cam usb_cam-test.launch
```

Sur le pc kemar a été installé les paquets ROS usb\_cam et usb\_cam\_node

Pour visualiser les images : roslaunch rviz rviz -> ajouter les topic /image

```
sudo apt-get install ros-indigo-usb_cam ros-indigo-image-view
```

## Connexion au robot

Se connecter au wifi « kemar\_wifi »

Comme le robot ne fait pas de dhcp, il faut spécifier les informations suivantes :

adresse IP : 192.168.11.XXX (tout sauf 100 qui est le robot et 1 qui est la passerelle)

netmask : 255.255.255.0

passerelle par défaut : 192.168.1.1

Pour que ROS marche, renseigner les variables d'environnement suivantes :

ROS\_HOSTNAME=192.168.11.XXX (avec l'adresse spécifiée au dessus)

ROS\_MASTER\_URI=http://192.168.11.100:11311

ROS\_IP=192.168.11.XXX (avec l'adresse spécifiée au dessus)

Sur le robot il faut modifier le fichier /etc/hosts et ajouter votre adresse IP et votre nom. Notez qu'une adresse est déjà enregistrée : 192.168.11.201

Sur votre pc dans /etc/hosts ajouter une ligne avec : 192.168.11.100 kemar\_base

## Installation des paquets Kemar sur un pc client

Extraire l'archive src-kemar.tar.gz dans le dossier workspace/src de l'environnement ROS (~/.catkin\_ws le plus souvent)

Installer les paquets suivants : ros-indigo-roboteq-\* et ros-indigo-joy

Executer catkin\_make

Dans le fichier kemar\_base.launch supprimer la ligne :

```
<include file="$(find kemar_robot)/launch/invert_odom.launch"/>
```

## Bouger le robot

Via ssh lancer sur le robot : roslaunch kemar\_robot kemar\_drive.launch

Pour bouger avec le clavier, sur le pc client lancer : roslaunch kemar\_teleop kemar\_keyboard.launch *kKeyboard*

Pour bouger avec la manette, sur le pc client lancer : roslaunch kemar\_teleop kemar\_joystick.launch *kJoy*

Avec la manette, il faut toujours appuyer sur le bouton LB (homme mort), utilisez le stick analogique gauche pour bouger et tourner.

### Installation de la manette xbox sur ROS :

sudo apt-get install ros-indigo-joy et brancher la manette

ls /dev/input et chercher jsX où X est le numéro du contrôleur xbox

sudo jstest /dev/input/jsX pour tester les touches

sudo chmod a+rw /dev/input/jsX

rosparam set joy\_node/dev « /dev/input/jsX »

roslaunch joy joy\_node

pour voir les touches appuyées : rostopic echo joy

*<param*

*bon manette*

*dans launchi*

## Créer une carte de navigation

Sur le pc client installer le paquet ros-indigo-map-server.

Sur le robot lancer roslaunch kemar\_robot kemar\_driver.launch pour allumer les moteurs

Sur le robot lancer roslaunch kemar\_robot kemar\_base.launch pour allumer le laser

Sur le client lancer roslaunch kemar\_navigation gmapping.launch

Faire bouger le robot dans l'environnement.

Pour visualiser la carte en direct, lancer sur le pc client : roslaunch rviz rviz et visualiser le topic /map

Pour sauvegarder la carte : roslaunch map\_server map\_saver -f nomfichier

Pour publier la carte sur le topic /map si besoin : `roslaunch map_server map_server nommap.yaml`

## Naviguer dans une carte

Attention, il peut y avoir des problèmes de synchronisation entre le pc kemar et le client si les calculs de trajectoire sont effectués sur ce dernier (grosse charge de calcul), il vaut mieux lancer les calculs sur le pc kemar.

Sur le pc robot lancer :

*/home/user/catkin\_ws/src/navigation\_seb/launch/*

Dans le fichier `move_base_digitalarti.launch` changer le nom de la carte ou la donner en paramètre

`roslaunch kemar_robot kemar_driver.launch` pour allumer les moteurs

*kMotor*

`roslaunch kemar_robot kemar_base.launch` pour allumer le laser

*kLaser*

`roslaunch navigation_seb move_base_digitalarti.launch` pour mapper les calculs au robot

*kCalcMap*

`roslaunch navigation_seb server_localisation.launch` pour calculer la position du robot dans la map

*kPosMap*

Sur le pc client lancer : `roslaunch navigation_seb view_navigation.launch`

*kView*

Initialiser la position du robot en cliquant sur le bouton « 2D Pose Estimate » puis en cliquant approximativement sur la position du robot dans la carte.

Pour donner un point d'arrivée au robot cliquer sur le bouton « 2D Nav Goal » puis cliquer sur un endroit de la carte.

## Problèmes connus

Attention, lorsque le robot est en arrêt d'urgence, couper la navigation et/ou la téléopération avant d'enlever l'arrêt d'urgence sinon le robot va partir à sa vitesse maximum !

*~ / bashrc*

*alias kemar="ssh user@19..."*

~~verif.~~  
*ligne 58 go*

*local. --*

*up : 5 → 25*  
*pub : 2.0 → 10*

*global  
update → 1 → 5*