

Communication entre un Arduino et ROS version Melodic

Le paquet ROS roserial utilise la communication de l'émetteur/récepteur asynchrone universel (UART) d'Arduino et convertit la carte en un nœud ROS qui peut publier des messages ROS et s'abonner à des messages également. L'éditeur du nœud Arduino ROS peut envoyer des données (provenant de capteurs ou de l'état du robot) de la carte à la machine exécutant ROS, tandis qu'un abonné au nœud Arduino ROS peut obtenir des instructions de la machine. La bibliothèque Arduino ros_lib permet à la carte Arduino de communiquer avec ROS.

Etapas communication entre ROS et Arduino :

Avant cette étape, vous devez installer arduino et ros sur Ubuntu.

[Installation/Windows - ROS Wiki](#)

[roserial arduino - ROS Wiki](#)

Sur Linux, pour installer arduino, écrivez dans un terminal : `sudo apt install arduino`

Pensez à sourcer votre environnement : `source (chemin du workspace)/devel/setup.bash`

1. Ensuite, vous devez copier et coller le code A dans votre terminal et appuyer sur ENTRÉE. Puis faire de même avec le code B.

A : `sudo apt install ros-melodic-roserial-arduino`

B : `sudo apt install ros-melodic-roserial`

Note : Il est possible que la clé d'activation ne soit plus bonne donc se référer à ce site pour la modifier (se référer à l'onglet solution) :

<https://answers.ros.org/question/325039/apt-update-fails-cannot-install-pkgs-key-not-working/>

2. Vous devez trouver le dossier arduino librairie dans votre HOME et ouvrir le terminal dessus : `cd ~/ (votre chemin)`

Par exemple pour moi : `cd ~/devel/ROS/src/braccio_arduino_ros_rviz/libraries`

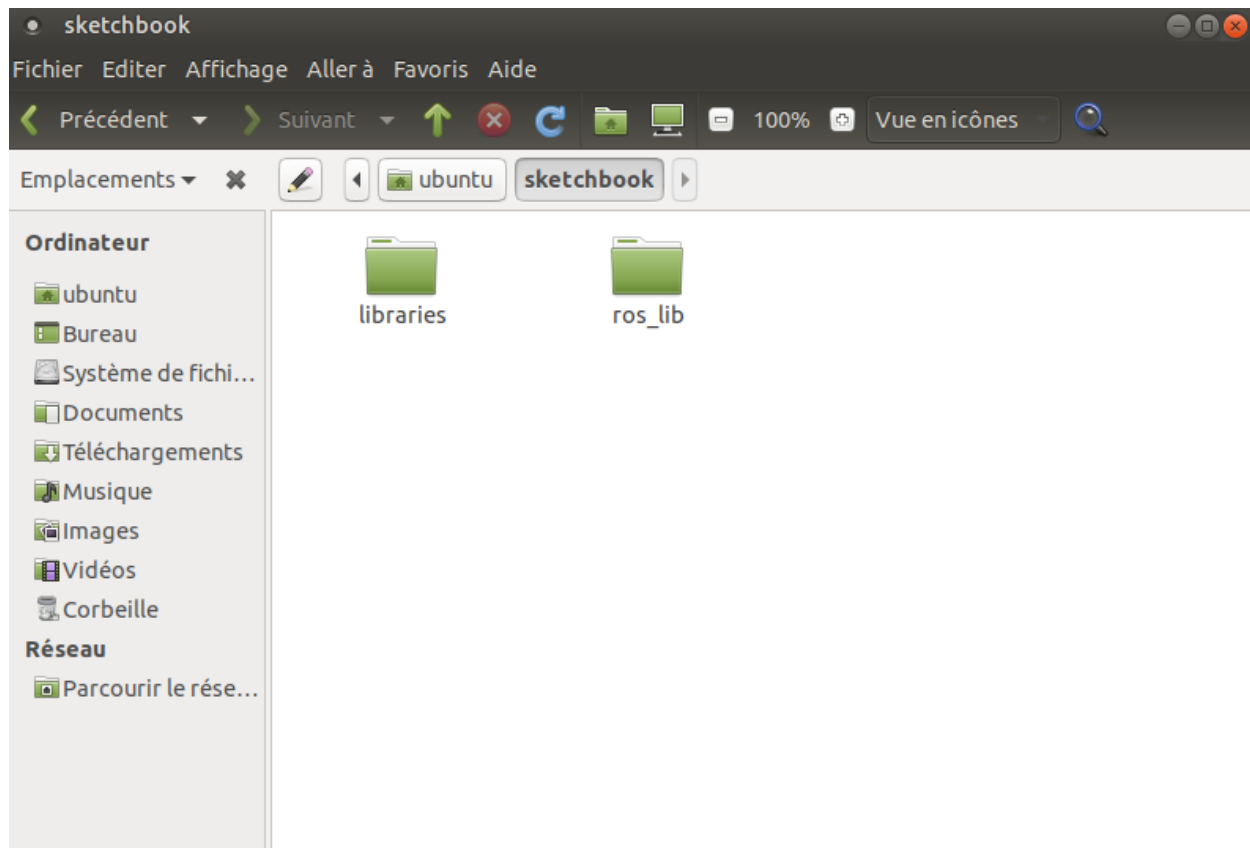
3. Vous devez copier/coller ce code dans ce terminal :

```
roslaunch roserial_arduino make_libraries.py
```

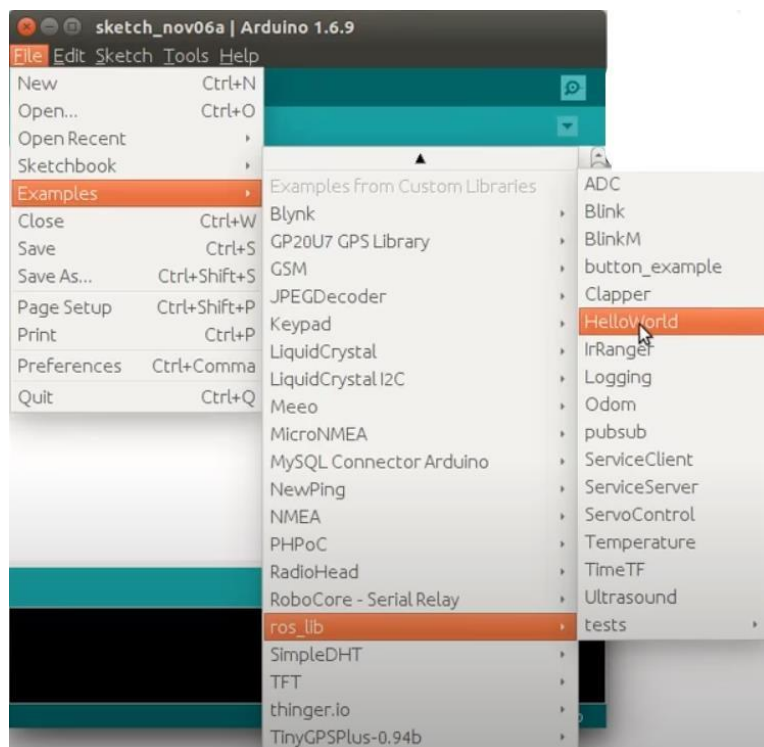
En cas d'erreur d'indication du sketchbook : Il faut ajouter votre chemin au code (la partie en rouge pour moi)

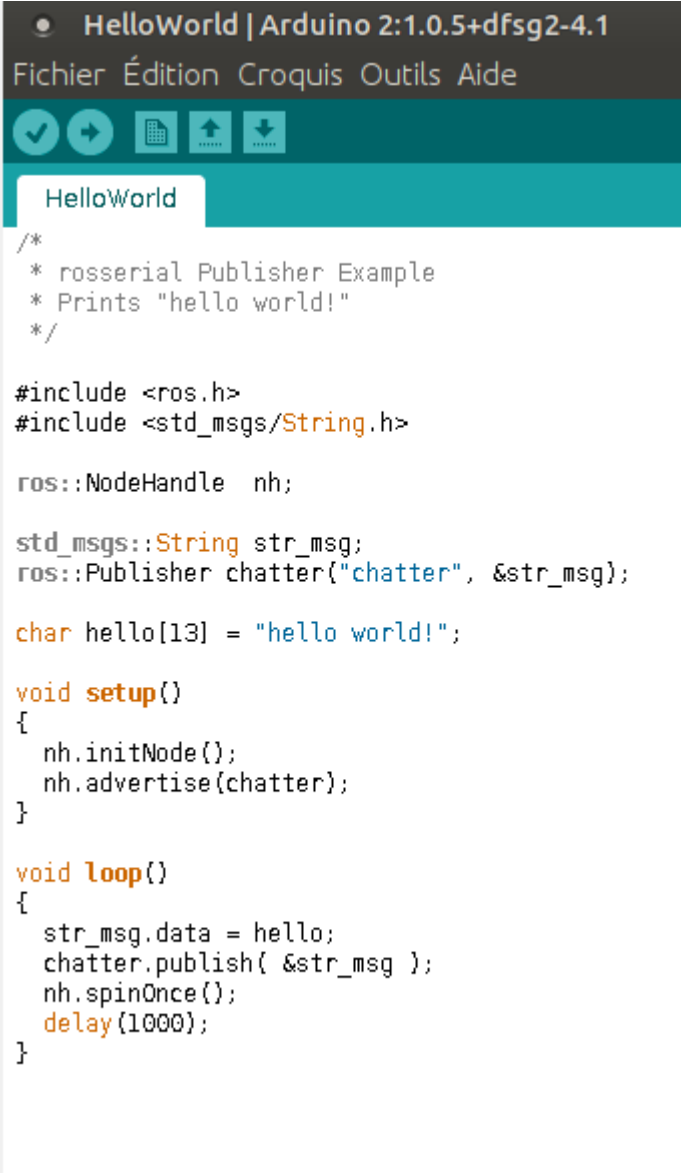
```
roslaunch roserial_arduino make_libraries.py /home/ubuntu/sketchbook/
```

Et appuyez sur Entrée. Un fichier ros_lib devrait apparaître dans votre sketchbook



4. Ouvrez ensuite l'IDE arduino et téléchargez un exemple de code dans votre carte.





```
/*
 * rosserial Publisher Example
 * Prints "hello world!"
 */

#include <ros.h>
#include <std_msgs/String.h>

ros::NodeHandle nh;

std_msgs::String str_msg;
ros::Publisher chatter("chatter", &str_msg);

char hello[13] = "hello world!";

void setup()
{
  nh.initNode();
  nh.advertise(chatter);
}

void loop()
{
  str_msg.data = hello;
  chatter.publish( &str_msg );
  nh.spinOnce();
  delay(1000);
}
```

Vous obtenez ce fichier

5. Maintenant voyons comment faire dans ros. Vérifiez si votre librairie ros arduino est installée ou non

Pour faire cette vérification il faut aller dans Home >catkin_ws>src>rosserial dans ce fichier vous devriez voir des fichiers rosserial en particulier le fichier rosserial_arduino et rosserial_msgs.

Pour ma part, j'ai vérifié cela en allant dans le fichier ros_lib qui a été créé suite à la commande ci-dessous :

```
roslaunch rosserial_arduino make_libraries.py /home/ubuntu/sketchbook/
```

Si cette vérification est bonne, Ouvrez votre terminal puis faire un catkin_ws

6. Démarrez le maître ros dans un autre terminal en utilisant la commande "roscore".

Connectez l'arduino avec ros :

```
roslaunch rosserial_python arduino_one.launch
```

Cette commande ne marche pas pour moi, j'ai une erreur :

```
ubuntu@ubuntu-VirtualBox:~/devel/ROS/src/braccio_arduino_ros_rviz/libraries$ ros
launch rosserial_python arduino_one.launch
RLError: [arduino_one.launch] is neither a launch file in package [rosserial
_python] nor is [rosserial_python] a launch file name
The traceback for the exception was written to the log file
ubuntu@ubuntu-VirtualBox:~/devel/ROS/src/braccio_arduino_ros_rviz/libraries$
```

Normalement si j'en crois les commandes à effectuer, celui-ci aurait du être installé au cours de l'installation de roserial au départ. Mais ça ne fonctionne pas et je ne trouve aucun paquet « roserial_python »

Peut-être y a-t-il encore des paquets à installer pour réussir à le faire marcher. Je n'ai pas trouvé la réponse.

Dans l'éditeur arduino le nom du noeud est chatter ... nous pouvons vérifier son travail ou non en utilisant "rostopic list".

Nous pouvons afficher le noeud chatter en utilisant "rostopic echo node_name".

Donc ici « rostopic echo chatter »

Vous devez changer la longueur des caractères dans le code arduino

Avant de télécharger votre code vers arduino, veuillez déconnecter votre connexion ros (master)

Maintenant le noeud publie les données de la carte arduino vers le ros

