

Arborescence Python pour CARLA et ROS

Afin de réaliser cette installation, on se basera sur un système d'exploitation Ubuntu 20.04 natif pour des raisons de compatibilité principalement.

Consultez la [documentation](#) et le [github](#) officiels pour des informations complémentaires sur CARLA-ROS-Bridge.

En cas de besoin, consultez cette [vidéo explicative](#) des commandes pouvant être utile à la résolution de problème.

Pour utiliser CARLA avec ROS, sur un système Ubuntu 20.04 vous devrez d'abord suivre ces procédures d'installations des logiciels :

- [Installation de ROS Noetic](#)*
- [Installation de CARLA-ROS Bridge](#)*

Ou alternativement sur Linux :

- [Installation de CARLA-ROS Bridge sur Docker](#)*

Pour plus d'informations sur l'utilisation ainsi que le fonctionnement de CARLA-ROS-Bridge, consultez la documentation suivante :

- [Utilisation de CARLA avec ROS](#)*
- [Fonctionnement théorique](#)*

Nous allons utiliser des scripts Python pour communiquer entre CARLA et ROS. En effet l'utilisation des scripts Python nous permet de créer des programmes flexibles pour interagir et traiter les données des capteurs dans ROS. Cela nous donne la liberté de personnaliser les fonctionnalités selon nos besoins, tout en profitant de la simplicité et de la puissance du langage Python.

Nous créerons ensuite un package catkin nous permettra d'exécuter un fichier Python pour communiquer avec un des éditeurs de notre nœud cible. Nous avons deux méthodes pour créer ces packages : « catkin_make » et « catkin build ». Nous utiliserons catkin build avec les commandes appropriées.

- Création d'un répertoire

```
mkdir -p ~/Documents/CARLA-ROS/Scripts/catkin_ws/src
```

- Installation des dépendances nécessaires

```
sudo apt install ros-noetic-catkin python3-catkin-tools
```

- Création du premier package

```
cd ~/Documents/CARLA-ROS/Scripts/catkin_ws && catkin build
```

- Configurez l'environnement du package

```
source ~/Documents/CARLA-ROS/Scripts/catkin_ws/devel/setup.bash
```

Il est recommandé d'ajouter cette ligne à la fin du fichier ~/.bashrc pour ne pas avoir à configurer l'environnement à chaque fois.

```
echo source ~/Documents/Stage-2024/CARLA-ROS/Scripts/Subscribers/catkin_ws/devel \>> ~/.bashrc
```

- Construction du package avec les dépendances

```
cd src/ && catkin_create_pkg sub_pkg rospy std_msgs sensor_msgs
```

Nous utilisons la commande `catkin_create_pkg` pour créer notre package avec dans notre cas ces dépendances nécessaires, telles que `rospy`, `std_msgs` et `sensor_msgs`.

Néanmoins, les dépendances peuvent être amenées à être différentes selon les scripts que vous voulez utiliser.

- Construction du sous-package avec les scripts

```
cd sub_pkg/ && catkin build
```

Une fois que notre arborescence est mise en place, nous plaçons nos fichiers Python dans le répertoire src/ de notre package.

Vous pouvez récupérer des scripts python en guise d'exemple [ici](#).

On obtient alors une arborescence comme celle-ci.

```
.
├── Scripts/
│   ├── catkin_ws /
│   │   ├── src /
│   │   │   ├── sub_pkg /
│   │   │   │   ├── src/
│   │   │   │   │   ├── python_script1.py
│   │   │   │   │   ├── python_script2.py
│   │   │   │   │   └── python_script3.py
```

- Donner les droits d'exécutions aux scripts

```
cd src/ && chmod +x *
```

- Exécuter un script python

```
roslaunch sub_pkg <PYTHON_FILE_NAME>.py
```

Voici un exemple de résultat de deux scripts python récupérant des données sur le véhicule dans CARLA (coordonnées géographiques, vitesse, etc.).

```
^Cgab@carla-ros-ns:~/Documents/Stage-2024/CARLA-R0S/Scripts/Subscribers/catkin_ws$ roslaunch sub_pkg subscribegnss.py
[INFO] [1714039427.017438, 137.192038]: Received NavSatFix data:
[INFO] [1714039427.018568, 137.192038]: Latitude: 0.000222
[INFO] [1714039427.019488, 137.192038]: Longitude: -0.000122
[INFO] [1714039427.020308, 137.192038]: Altitude: 2.119429
gab@carla-ros-ns:~/Documents/Stage-2024/CARLA-R0S/Scripts/Subscribers/catkin_ws$ $ roslaunch sub_pkg subscribeimy.py
[INFO] [1714039431.042388, 138.542038]: Received Imu data:
[INFO] [1714039431.043509, 138.542038]: Linear Acceleration: x=0.496962, y=-0.002661, z=9.797404
[INFO] [1714039431.044428, 138.542038]: Angular Velocity: x=-0.000000, y=0.000000, z=-0.000000
[INFO] [1714039431.045319, 138.542038]: Orientation: x=-0.025308, y=-0.001229, z=-0.998228, w=0.053846
```

- Reconstruire le package après une modification

```
cd ~/Documents/CARLA-ROS/Scripts/catkin_ws && catkin build
```

*Lorsque nous effectuons des modifications ou ajoutons des fichiers Python dans le répertoire « **sub_pkg** », nous devons nous assurer de reconstruire l'ensemble du package situé dans « **catkin_ws** » pour que les changements soient pris en compte.*

- Scripts d'automatisation

Pour simplifier ces étapes, n'hésitez pas à utiliser ces différents scripts BASH pour automatiser ces étapes :

- [Construire l'arborescence](#)
- [Ajoutez ou des scripts python à l'arborescence](#)
- [Lancer un fichier python](#)