1/ Préparer son espace de travail.

Si possible, il faudrait que tout le monde ait un workspace. (J'ai vérifié la mémoire normalement, cela devrait passer.)

```
bash
mkdir ros2_name
cd ros2_jname
```

2/ Builder son espace de travail.

Il faut builder votre workspace vide sans rien pour bien le paramétrer. N'oubliez pas de sourcer le workspace.

```
colcon build
source ~/ros2_name/install/local_setup.bash
source /opt/ros/humble/setup.bash
```

Une fois votre workspace buildé, normalement des fichiers se sont générés : build, install et log. Cela signifie que le build s'est bien passé.

3/ Transfert du fichier src.

Localement sur votre PC, vous avez votre code que vous souhaitez exécuter, notamment dans le fichier source. Le fonctionnement est toujours le même dans ROS, où tous les packages sont dans le dossier "src".

Vous pouvez copier directement le code localement de votre PC vers la Raspberry Pi avec la commande suivante. Pour cela, je me place dans mon terminal dans mon ros2_ws de mon PC (attention, je fais cela sur mon PC, pas en SSH), puis j'exécute cette commande. Remplacez "ros2_johann" par "ros2_raphael", par exemple.

```
bash
scp -r src pi@10.105.1.168:~/ros2_name/src/ (mettre @IP voiture)
```

Vérifiez que dans votre terminal connecté en SSH, le répertoire "src" a bien été créé. Ensuite,

refaites un colcon build. Normalement, cela fonctionne, et vous pouvez exécuter votre code. Si vous avez des modifications, vous pouvez renvoyer le code de la Raspberry Pi vers votre PC en faisant simplement la commande inverse.

colcon build
source ~/ros2_name/install/local_setup.bash
source /opt/ros/humble/setup.bash

Attention : "colcon build" doit toujours être exécuté dans votre workspace.

Figure : Exemple d'un workspace sur ROS.