# Configuration d’une Raspberry Pi 4 avec Ubuntu et Environnement de Développement Qt

Créateur **Benjamin Bachelard** Créé **Jan 27, 2025, 16:48** Dernière mise à jour **Jan 30, 2025, 11:03**

## Étape 1 : Installation d’Ubuntu sur la Raspberry Pi

### 1.1. Téléchargement de l’image Ubuntu

1. Accédez à Ubuntu Raspberry Pi Downloads.
2. Téléchargez **Ubuntu Server 22.04 LTS** (recommandé).

### 1.2. Flashage de la carte SD

1. Utilisez **Raspberry Pi Imager** ou **Balena Etcher** pour flasher l’image Ubuntu sur la carte SD.
2. Sélectionnez l’image Ubuntu téléchargée et choisissez la carte SD comme destination.
3. Lancez le flashage.

### 1.3. Premier démarrage de la Raspberry Pi

1. Insérez la carte SD dans la Raspberry Pi et allumez-la.
2. Connectez-vous avec les identifiants par défaut :
    - **Utilisateur** : `ubuntu`
    - **Mot de passe** : ruche1234

### 1.4. Activer SSH dès l'installation

1. Créez un fichier vide nommé `ssh` dans la partition `boot` de la carte SD :

```
touch /boot/ssh
```

### 1.5. Premier démarrage de la Raspberry Pi

1. Insérez la carte SD dans la Raspberry Pi et allumez-la.
2. Connectez-vous via SSH depuis un autre PC :

```
ssh ubuntu@172.21.28.23
```

1. les identifiants par défaut :
    - **Utilisateur** : `ubuntu`
    - **Mot de passe** : ruche1234

## Étape 2 : Mise à jour du système

1. Mettez à jour les paquets et le système :

```
sudo apt update && sudo apt upgrade -y
```

1. Redémarrez pour appliquer les changements :

```
sudo reboot
```

# Étape 3 : Installation des outils de base

## 3.1. Installer Apache pour héberger un serveur web

1. Installez Apache :

```
sudo apt install apache2 -y
```

1. Activez et démarrez Apache :

```
sudo systemctl enable apache2
```

```
sudo systemctl start apache2
```

1. Testez en accédant à `http://<adresse-ip>` depuis un navigateur.

## 3.2. Installer MariaDB (optionnel)

1. Installez le serveur de base de données :

```
sudo apt install mariadb-server -y
```

1. Activez et démarrez MariaDB :

```
sudo systemctl enable mariadb
```

```
sudo systemctl start mariadb
```

## 3.3. Installer PHP (optionnel)

1. Installez PHP pour les scripts dynamiques :

```
sudo apt install php libapache2-mod-php -y
```

# Étape 4 : Installation de Qt et de l'environnement de développement

## 4.1. Installer Qt sur la Raspberry Pi

1. J'installe les outils de base Qt :

```
sudo apt install qtbase5-dev qt5-qmake g++ make -y
```

1. J'installe OpenCV pour la gestion des images :

```
sudo apt install libopencv-dev python3-opencv -y
```

## 4.2. Préparer la cross-compilation avec Qt

## Sur la Raspberry Pi (cible) :

1. J'installe les outils nécessaires :

```
sudo apt install rsync build-essential gdb -y
```

1. Je copie les fichiers système requis vers le PC hôte :

```
rsync -avz --delete /lib ubuntu@<adresse-ip-de-la-raspberry>:~/rpi-sysroot/
```

rsync -avz --delete /usr ubuntu@<adresse-ip-de-la-raspberry>:~/rpi-sysroot/

esync -avz --delete /opt ubuntu@<adresse-ip-de-la-raspberry>:~/rpi-sysroot/

## Sur le PC (hôte) :

1. J'installe Qt Creator et le compilateur cross-compile :

sudo apt install qtcreator g++-arm-linux-gnueabihf -y

1. Je configure Qt Creator :

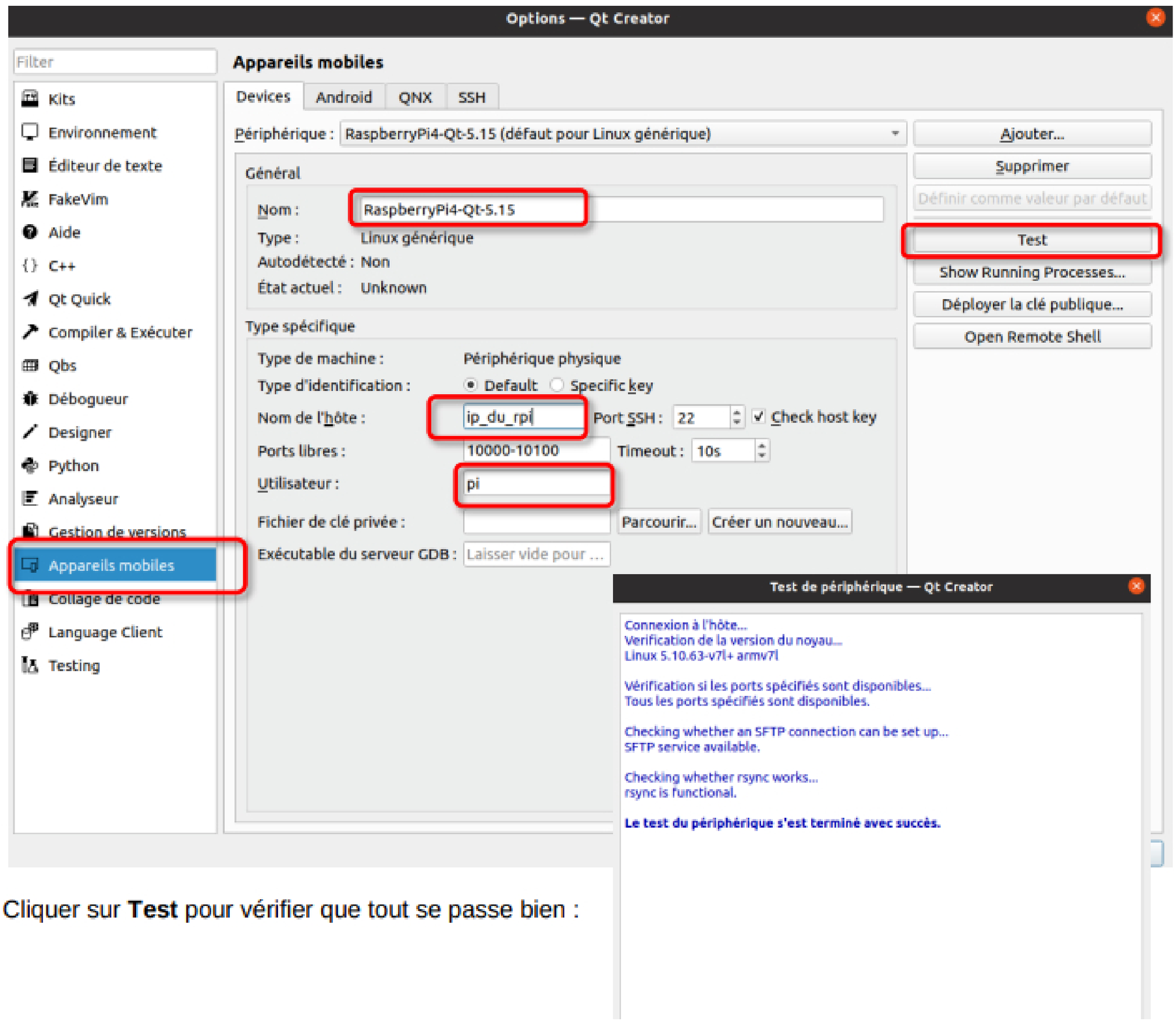
- Je crée un nouveau kit de compilation avec :
  - Compilateur : `arm-linux-gnueabihf-gcc`
  - Sysroot : les fichiers copiés depuis la Raspberry Pi.
  - Je teste la compilation avec un projet simple.

Ensuite Sur QT:

## 5 Configuration de Qt Creator pour la Cross-compilation

Aller dans le menu **Outils – Options…**

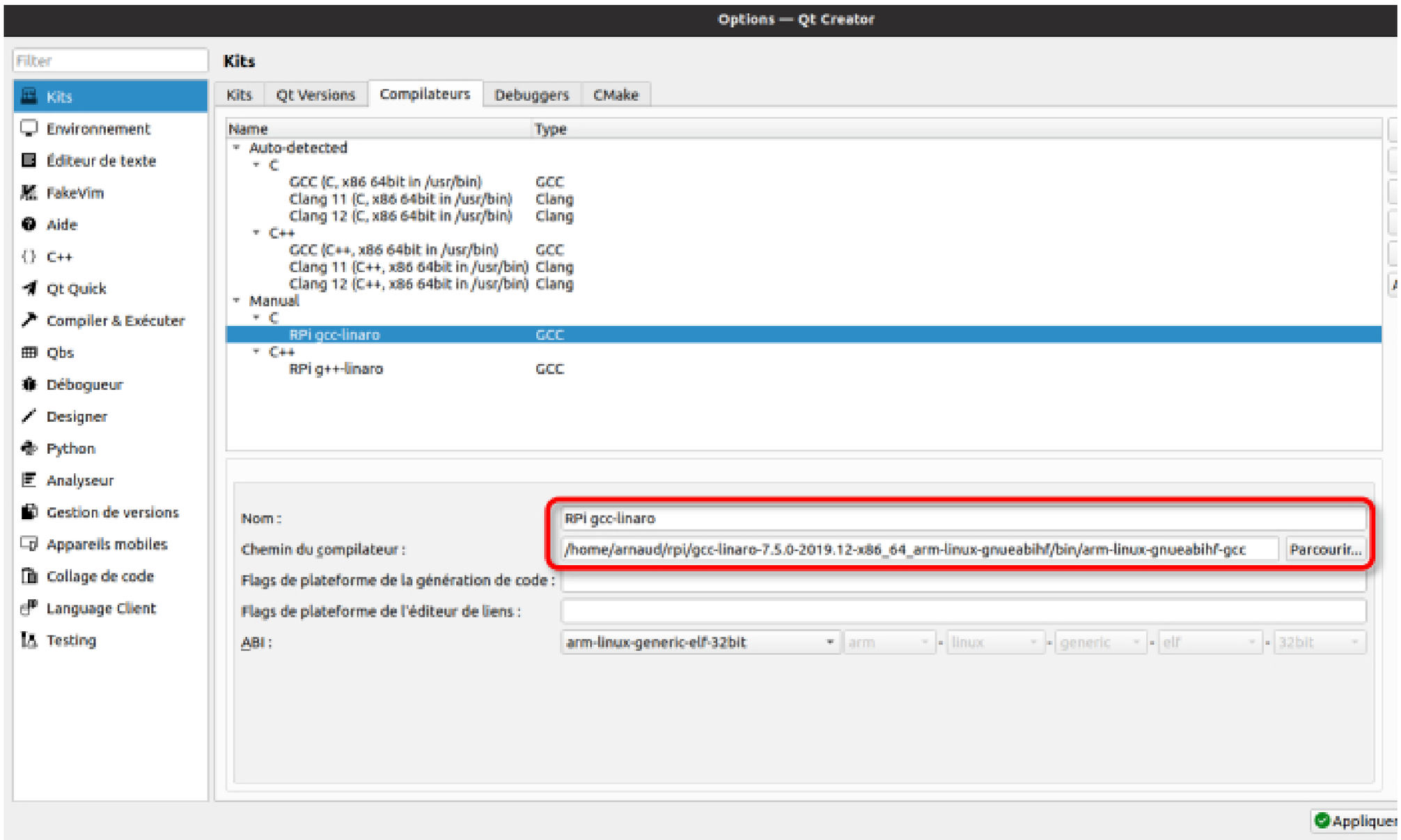
On va commencer par créer un nouvel appareil, cliquer sur **Appareils mobiles** et renseigner comme ci-dessous :




Cliquer sur **Test** pour vérifier que tout se passe bien :

Aller ensuite dans **Kits** :

Dans l'onglet **Compilateurs**, renseigner le chemin des compilateurs C et C++ :




Pour le compilateur C :

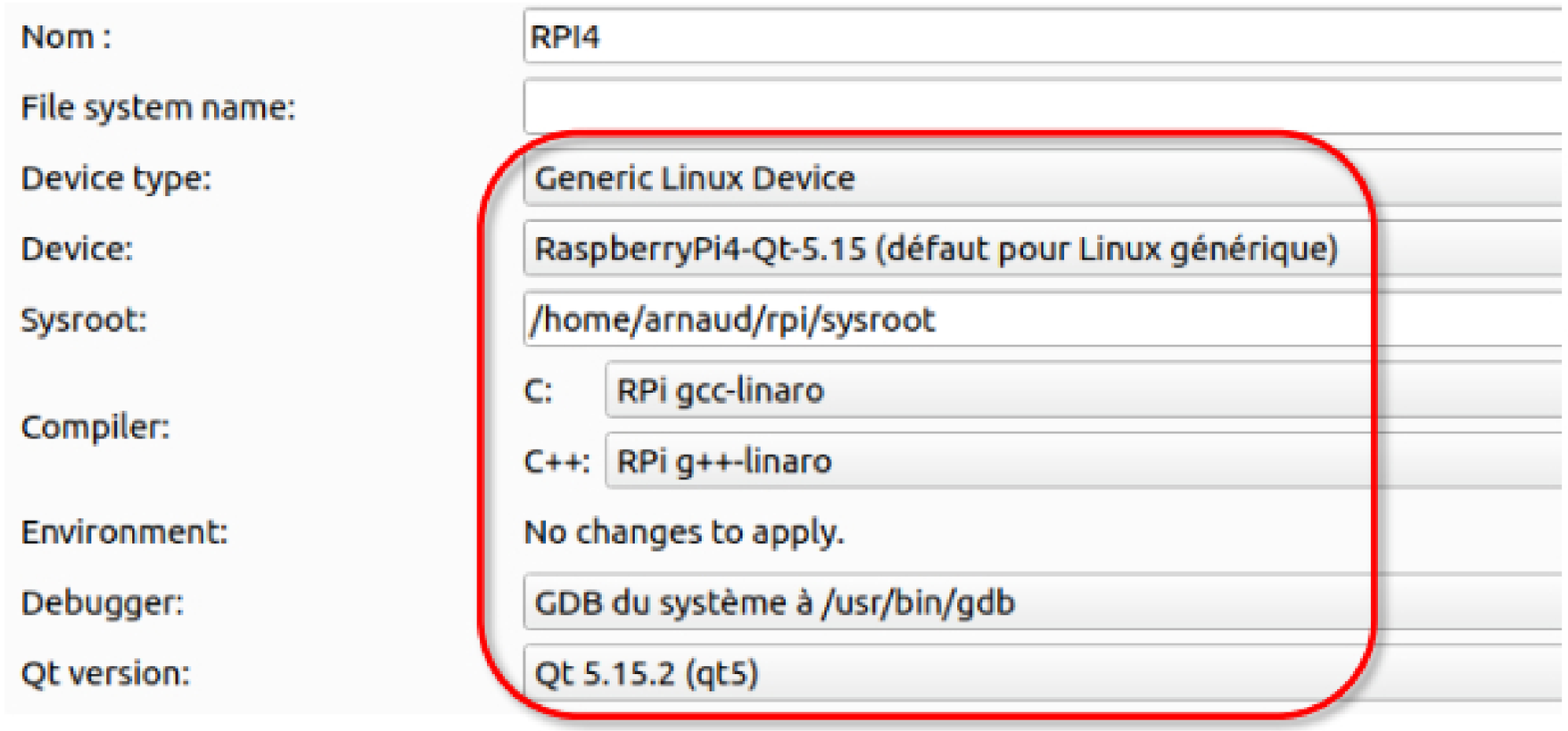
```
/home/user/rpi/gcc-linaro-7.5.0-2019.12-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-gcc
```

Pour le compilateur C++ :

```
/home/user/rpi/gcc-linaro-7.5.0-2019.12-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-g+
```

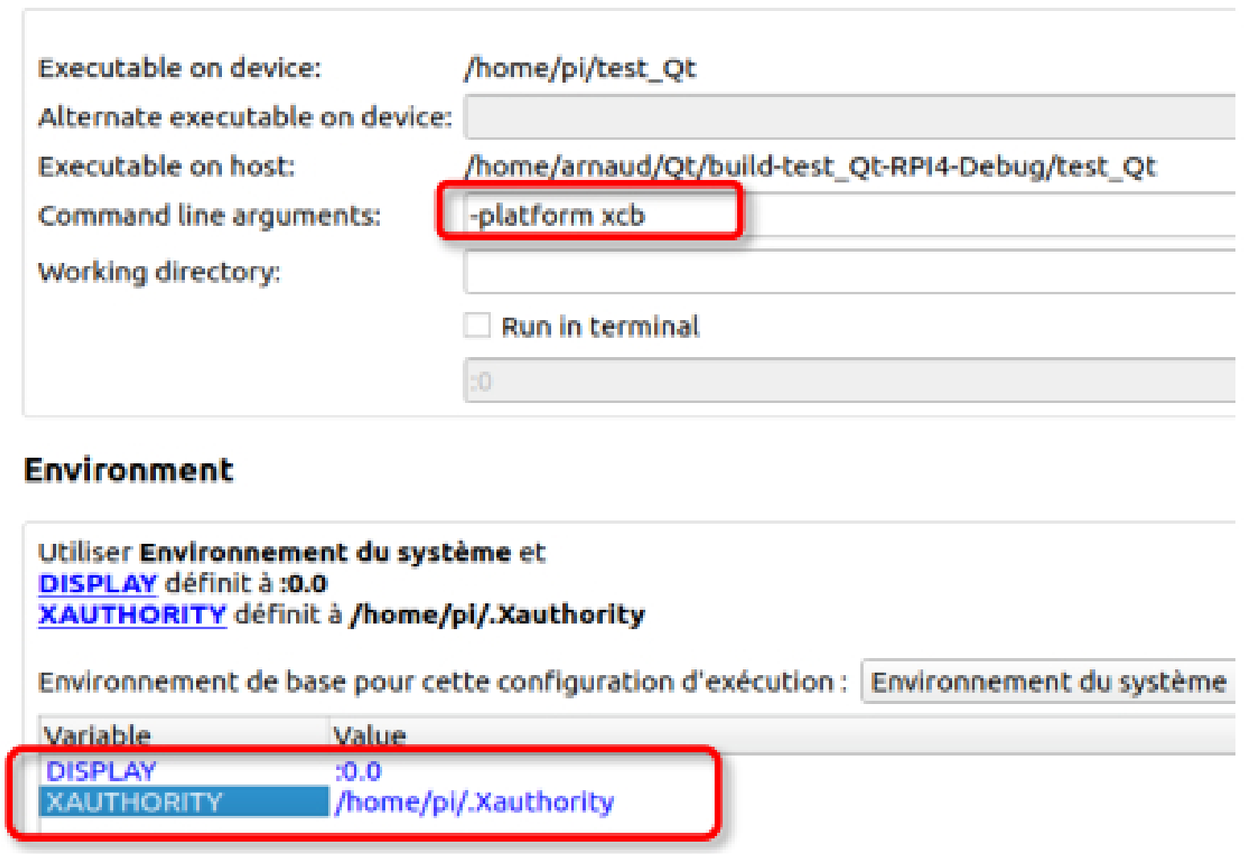
Dans l'onglet **Qt Versions** , renseigner l'emplacement du fichier **qmake** :

```
/home/user/rpi4/qt5/bin
```




Valider vos changements.

Avant de compiler, il faut signifier à votre programme Qt de s'afficher sur l'écran principal de la carte. Dans **Projet** puis **Run** on ajoute :




En cas d'utilisation d'un écran LCD on pourra indiquer sa taille dans les variables d'Environnement :

QT_QPA_EGLFS_PHYSICAL_WIDTH=211

QT_QPA_EGLFS_PHYSICAL_HEIGHT=127