

Université Tunis El-Manar	Année Universitaire : 2021-2022
Faculté des Sciences de Tunis	Module : CIOC
Section : IF5 Option IOT	Enseignant : C.A. ABID

TP1 Préparation de l'environnement de développement

Partie I : Installation de Raspberry OS Lite et connexion distante SSH

On se propose d'installer la dernière version de la distribution officielle Raspberry OS. La système sera installé sur une carte SD.

- Télécharger le fichier image du système d'exploitation à installer sur la Raspberry Pi à partir de <https://www.raspberrypi.org/software/operating-systems/#raspberrypi-os-32-bit>

Veillez choisir Raspberry OS Lite.

- Placer la carte SD dans le lecteur de cartes, puis à l'aide de la commande `lsblk`, déterminer le nom des partitions de la carte SD

- Il faut démonter les partitions présentes sur la carte en utilisant la commande `umount`. Par exemple : `umount /dev/sdX1` avec `/dev/sdX1` est une partition de la carte `/dev/sdX`.

- Maintenant, nous copions les fichiers de la distribution pour le Raspberry Pi sur la carte SD. Nous utilisons la commande `dd` pour reproduire des zones de disque qui ne font pas partie d'un système de fichier : secteur de démarrage (le MBR), tables de partition, etc.

```
dd bs=4M if=2021-03-04-raspios-buster-armhf-lite.img  
of=/dev/sdX status=progress conv=fsync
```

L'option `bs` permet de spécifier la taille de chaque bloc. L'option `if` permet de préciser le chemin du fichier image du système d'exploitation à installer. L'option `of` spécifie le nom de la carte SD sur laquelle le système d'exploitation va être créé. Il faut bien rentrer le nom de la carte et pas d'une partition, c'est à dire enlever le chiffre au bout du nom des partitions (par exemple `sdc`, pas `sdc1` ou `mmcblk0` mais pas `mmcblk0p1`).

- Une fois la création du système sur la carte est effectuée, nous continuons par configurer la connexion au réseau Wifi et l'activation du serveur SSH. Cela nous permettra de se connecter à la carte à travers le protocole SSH.

Dans la première partition, dans le répertoire `/boot` :

- Créer un fichier `wpa_supplicant.conf` contenant le code suivant :

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev  
update_config=1  
country=FR
```

```
network={
ssid="SSID"
psk="password"
key_mgmt=WPA-PSK
}
```

Veillez remplacer le SSID et password par les paramètres du réseau Wifi auquel vous êtes connectés.

- Créer un fichier vide appelé ssh. Ce fichier permet d'indiquer à l'OS que le protocole SSH doit être activé.
- Lancez la commande ci-après afin de vider le cache d'écriture et permettre l'éjection sans risque de la carte SD :

```
sudo sync
```

- Quand la commande vous rend la main, sortez la carte du lecteur de votre ordinateur et l'insérez la dans la Raspberry Pi.

Normalement, si la configuration est effectuée correctement, après être connectée à l'alimentation, la carte Raspberry Pi démarrera avec la distribution Raspberry OS, tout en autorisant la connexion au système via le protocole SSH avec le login pi et le mot de passe par défaut raspberry. Nous devons pour cela retrouver l'adresse IP de la carte sur le réseau à l'aide de la commande nmap :

```
sudo nmap -sS -p 22 x.x.x.0/24
```

Une fois l'adresse IP est identifiée , vous pouvez vous connecter à la carte à travers le protocole SSH :

```
$ ssh pi@<adresse IP de la carte>
```

Partie II : Configuration d'Eclipse pour la compilation croisée

- Sur la RPi

- Installer la bibliothèque pigpio
wget <https://github.com/joan2937/pigpio/archive/master.zip>
unzip master.zip
cd pigpio-master
make
sudo make install
S'il y a un problème à cause de python, il faut installer les paquets suivants :
sudo apt install python-setuptools python3-setuptools
- Installer gdbserver pour les opérations de débogage distantes
sudo apt install gdbserver

- Sur le PC

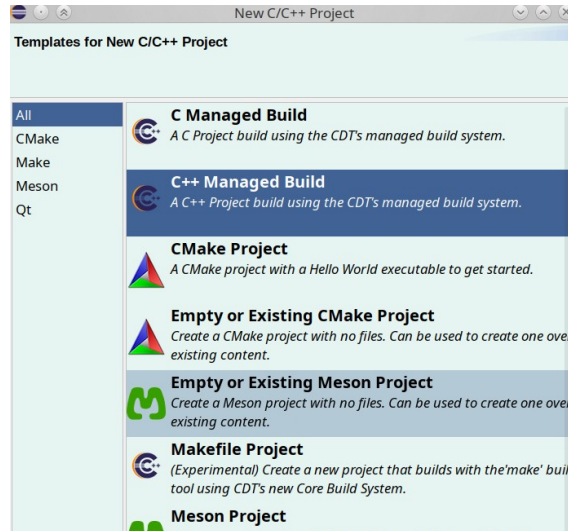
- Télécharger et installer 'Eclipse IDE for C++ Developers' sur Ubuntu :
<https://www.eclipse.org/downloads/packages/release/2021-03/r/eclipse-ide-cc-developers>
- Télécharger la chaîne de compilation croisée pour la Rpi
<https://github.com/Pro/raspi-toolchain/releases/latest/download/raspi-toolchain.tar.gz>
Installer la chaîne de compilation dans le répertoire /opt/cross-pi-gcc
sudo tar xzf raspi-toolchain.tar.gz --strip-components=1 -C /opt
- Installer gdb-multiarch pour le débogage à distance
sudo apt install gdb-multiarch
- Créer une image des bibliothèques installées sur la carte Rpi (cette opération nécessite quelques heures pour s'exécuter)

```
rsync -vR --progress -rl --delete-after --safe-links  
pi@192.168.1.Pi:/{lib,usr,etc/ld.so.conf.d,opt/vc/lib} $HOME/rootfs
```

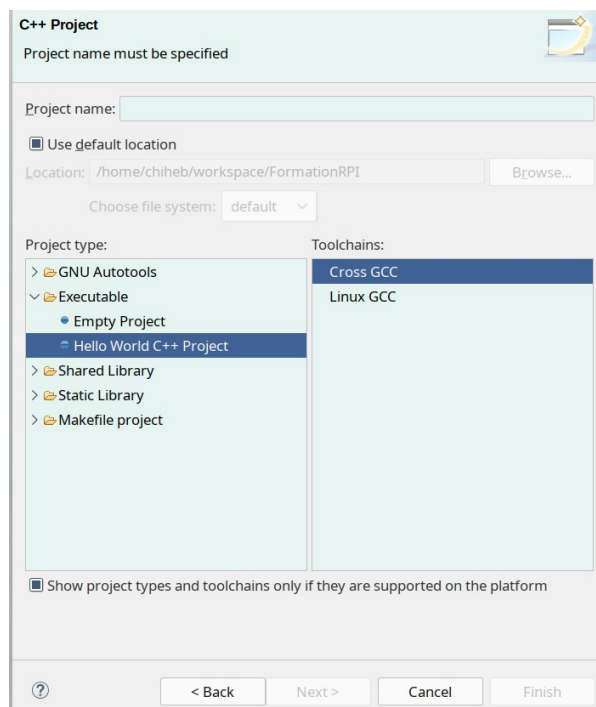
De cette façon, dans le répertoire \$HOME/rootfs on aura toutes les bibliothèques déjà installées sur la carte Rpi, ce qui nous permettra d'effectuer la compilation des programmes qui utilisent ces bibliothèques.

Compilation et débogage avec Eclipse IDE

- Pour la création d'un nouveau projet, aller à File→New → C/C++ Project
- Choisir C++ Managed Build

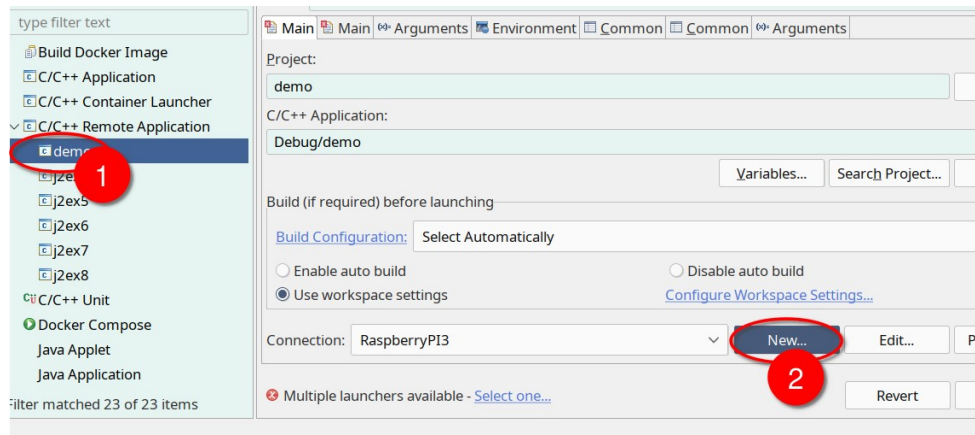


- Choisir Hello World C++ Project et préciser le nom du projet 'demo1'

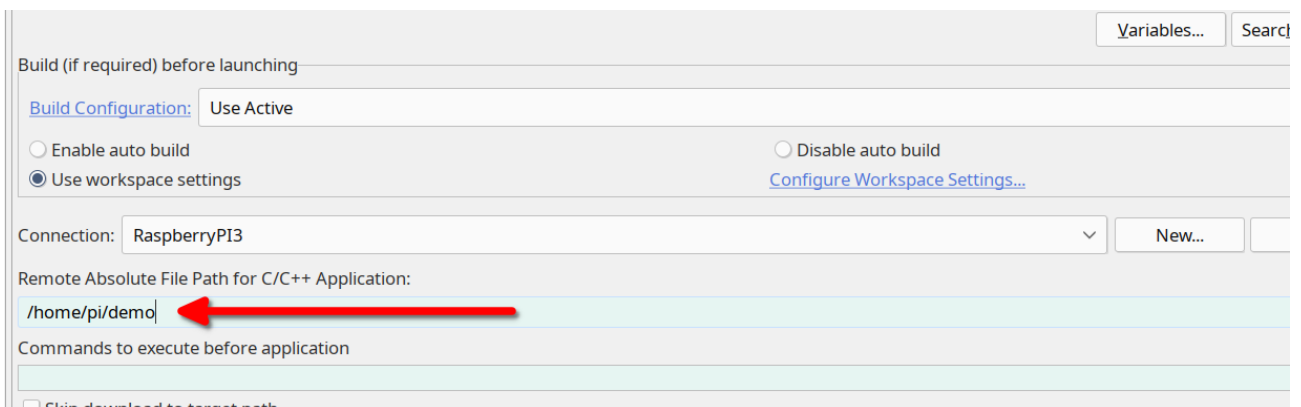


- Dans la dernière fenêtre, préciser arm-linux-gnueabihf- comme préfixe et /opt/cross-pi-gcc/bin comme chemin de la chaîne de compilation
- Vérifier que la compilation s'effectue correctement

- Pour l'exécution et le déploiement de votre programme sur la carte Rpi, choisir 'Run → Run configuration'. Puis, créer une nouvelle configuration de type 'C/C++ Remote Application'. Ensuite, créer une nouvelle connexion SSH pour réaliser le déploiement automatique sur la carte RPi.



- Préciser le chemin absolu du programme distant à exécuter. Le bouton 'Run' s'activera et il est possible par la suite de déployer le programme et l'exécuter automatiquement.



- Pour lancer une session de débogage, choisir Run → Debug Configuration. Puis, choisir l'onglet Debugger. Préciser le débogueur gdb-multiarch pour le champ 'GDB debugger'

