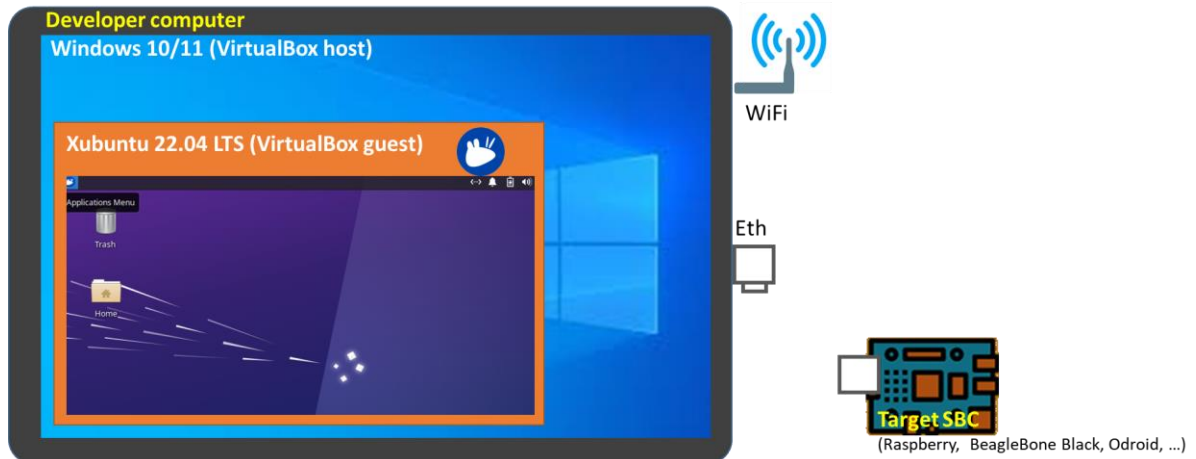


## Contenido

1	Elementos y nomenclatura .....	2
1.1	Configuración del SBC .....	2
2	Pasos para instalación.....	2
2.1	Instalaciones y descargas .....	2
2.2	Instalar máquina virtual pre-instalada .....	2
2.3	Preparar las conexiones de red de la máquina VirtualBox .....	3
2.3.1	Configuración de red 1 .....	3
2.3.2	Configuración de red 2 .....	4
2.3.3	Configuración de red 3 .....	4
2.4	Arrancar la máquina VBGuest .....	4
2.4.1	Establecer configuración de red en máquina VBGuest .....	4
2.5	Conectividad VBGuest $\leftrightarrow$ BBB.....	5
2.5.1	Control de BBB con SSH .....	5
2.5.2	Copia de archivos a/desde BBB con scp .....	5
2.5.3	Comprobar puertos serie en BBB .....	5
2.5.4	Bridge para acceso a internet desde el SBC .....	6
3	Uso de kit BBB en Qt Creator (programas modo consola) .....	6
3.1	Con qmake .....	6
3.2	Seleccionar el kit a aplicar:.....	7
3.3	Modificar conexión de dispositivo (si es necesario) .....	7
3.4	Instalar aplicación para ejecución en el arranque de la BBB .....	8
3.4.1	Crear archivo con script de arranque .....	8
3.4.2	Crear archivo de servicio .....	9
3.4.3	Copiar archivos a BBB.....	9
3.4.4	Instalar servicio en BBB .....	9
3.4.5	Desinstalar servicio en BBB .....	9

# 1 Elementos y nomenclatura

- **Target SBC (SBC):** equipo destino del programa a desarrollar (BeagleBone Black, Raspberry Pi, etc.), que se desea funcione de manera autónoma.
- **Developer computer/VirtualBox Host (VBHost):** equipo de desarrollo del usuario, típicamente corriendo alguna versión moderna de Windows (10/11).
- **Virtual Machine/VirtualBox Guest (VBGuest):** máquina virtual Linux XUbuntu 22.04 que se ejecuta en el Host. Utiliza la técnica de virtualización para acceder a los recursos hardware del host (disco, memoria, comunicaciones). En ella se desarrollarán los programas para el Target SBC.



## 1.1 Configuración del SBC

Por defecto, el **SBC** utilizado (BeagleBone Black-**BBB**) tiene pre-instalado un Linux Debian 11 con una dirección IP fija para su único interfaz de red (cableada 10/100 Mbps). Los datos fundamentales para el uso de este equipo son:

Usuario: debian

Clave: tempwd

Dirección IP pre-asignada: 192.168.100.23

Máscara de subred: 255.255.255.0

Gateway: 192.168.100.100

## 2 Pasos para instalación

### 2.1 Instalaciones y descargas

Descargar e instalar VirtualBox desde: <https://www.oracle.com/es/virtualization/virtualbox/>

Descargar desde un navegador el archivo zip con la máquina virtual: [https://drive.google.com/file/d/1O-sp57\\_L6b4OofSJnm4S\\_BMhJKH-otH/view?usp=drive\\_link](https://drive.google.com/file/d/1O-sp57_L6b4OofSJnm4S_BMhJKH-otH/view?usp=drive_link), y descomprimir en la carpeta deseada.

### 2.2 Instalar máquina virtual pre-instalada

Abrir VirtualBox y añadir máquina virtual, con la configuración mínima siguiente:

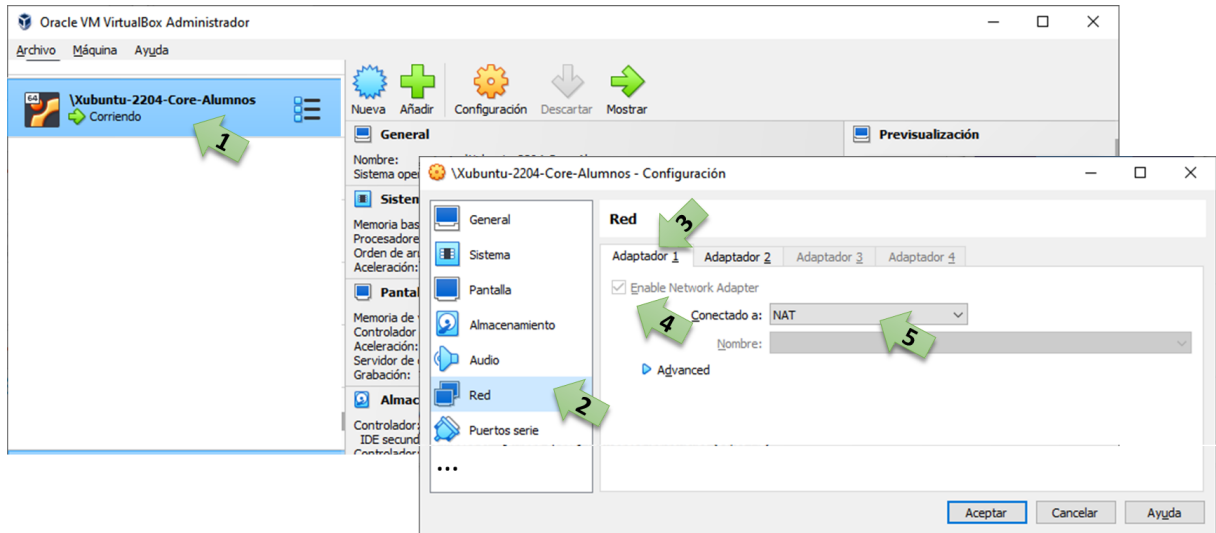
- Name and Operating System
  - Name: XUbuntu-2204-Core-Alumnos
  - Type: Linux XUbuntu 64-bit
- Hardware:
  - Base memory: 2048Mb Ram (4096 recomendado)
  - Processors: 1 (2 recomendado)
- Hard Disk:
  - Use an existing Virtual Hard Disk File → Seleccionar el archivo descomprimido previamente `_Xubuntu-2204-Core-Alumnos.vdi`

A continuación se podrán modificar otras características hardware, como las conexiones de red. Se verá en los puntos posteriores.

### 2.3 Preparar las conexiones de red de la máquina VirtualBox

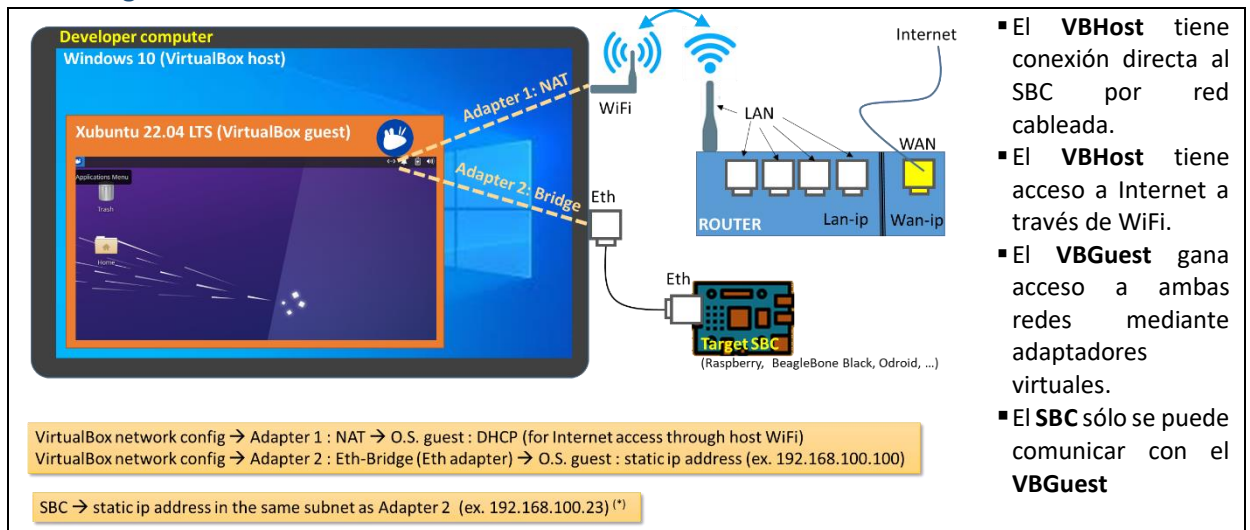
Dependiendo de la configuración de sus redes y conexiones, habrá que establecer en la configuración de red del **SBC**, así como la asignación de recursos virtuales del **VBGuest** para la comunicación.

Para ello, antes de arrancar la máquina virtual, habrá que modificar su configuración de red desde VirtualBox.

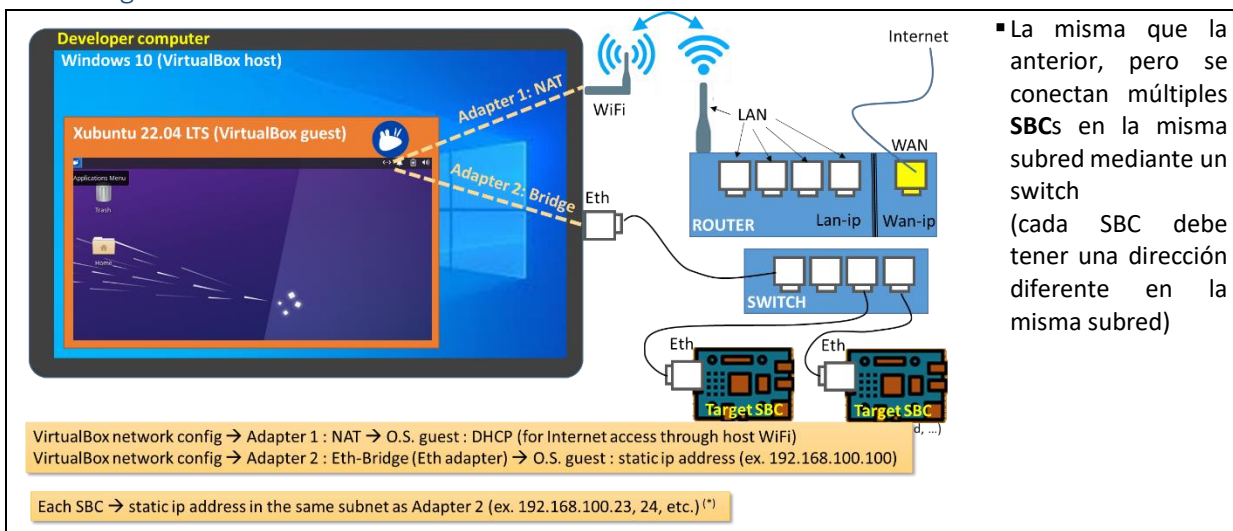


Algunos ejemplos:

#### 2.3.1 Configuración de red 1

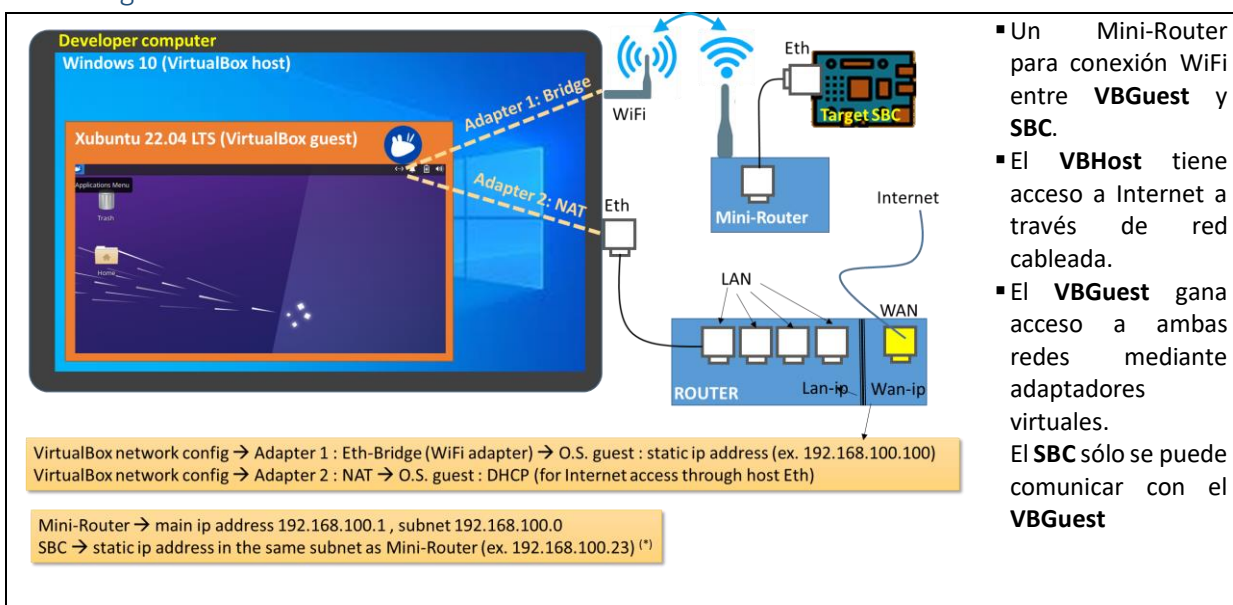


### 2.3.2 Configuración de red 2



- La misma que la anterior, pero se conectan múltiples SBCs en la misma subred mediante un switch (cada SBC debe tener una dirección diferente en la misma subred)

### 2.3.3 Configuración de red 3



- Un Mini-Router para conexión WiFi entre VBGuest y SBC.
- El VBHost tiene acceso a Internet a través de red cableada.
- El VBGuest gana acceso a ambas redes mediante adaptadores virtuales. El SBC sólo se puede comunicar con el VBGuest

## 2.4 Arrancar la máquina VBGuest

Arrancar la máquina virtual desde VirtualBox, hasta su arranque completo. Presentará un login donde los datos a introducir son:

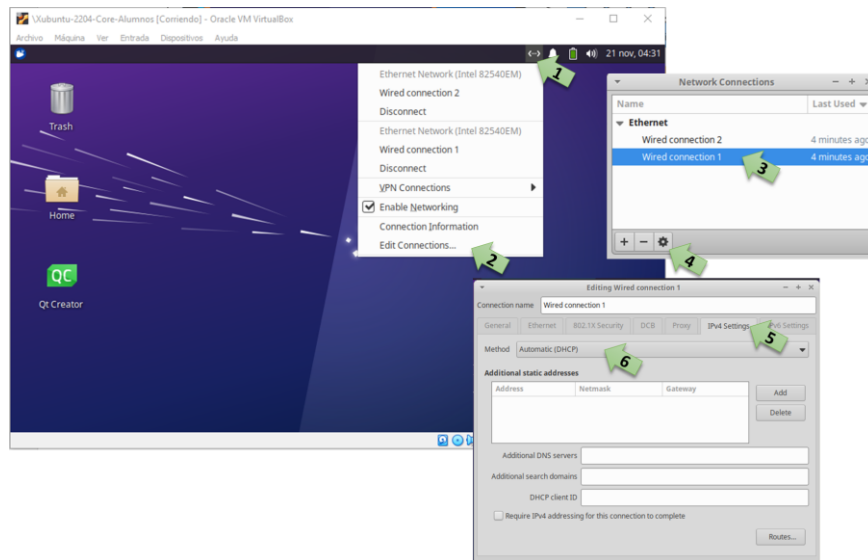
User: alumno

Password: isauniovi2022

### 2.4.1 Establecer configuración de red en máquina VBGuest

La configuración a establecer depende de los settings necesarios para VBGuest del punto 2.3. Se realizan mediante los pasos siguientes para el adaptador 1 y el adaptador 2.

- Para un adaptador que use DHCP, simplemente hay que seleccionar esa opción en "IPv4 Settings".
- Para un adaptador con dirección fija, se selecciona método Manual y se añaden los datos de la dirección IP:
  - Address: 192.160.100.100
  - NetMask: 24
  - GateWay: vacío



## 2.5 Conectividad VBGuest ↔ BBB

Conectar la BBB a la red según la configuración que se haya elegido.

Ejecutar la consola en VBGuest. Comprobar conectividad con ping:

```
alumno@alumno-VirtualBox:~$ ping 192.168.100.23
PING 192.168.100.23 (192.168.100.23) 56(84) bytes of data.
64 bytes from 192.168.100.23: icmp_seq=1 ttl=64 time=2.33 ms
64 bytes from 192.168.100.23: icmp_seq=2 ttl=64 time=1.96 ms
64 bytes from 192.168.100.23: icmp_seq=3 ttl=64 time=2.43 ms
64 bytes from 192.168.100.23: icmp_seq=4 ttl=64 time=2.26 ms
Terminar con Ctrl-C
```

### 2.5.1 Control de BBB con SSH

Conectar mediante SSH:

```
alumno@alumno-VirtualBox:~$ ssh debian@192.168.100.23
Password: temppwd
...
debian@beaglebone:~$
```

Los siguientes comandos que se introduzcan se ejecutarán en la BBB. Para terminar:

```
debian@beaglebone:~$ exit
alumno@alumno-VirtualBox:~$
```

La conexión puede dar un error relativo a las claves criptográficas guardadas, que dependen de cada BBB conectada. En caso de obtener este error, eliminar los archivos siguientes y proceder con el paso 3.3 antes de reintentar:

```
alumno@alumno-VirtualBox:~$ rm /home/alumno/.ssh/known_hosts
alumno@alumno-VirtualBox:~$ rm /home/alumno/.ssh/qtc_bbb*
```

### 2.5.2 Copia de archivos a/desde BBB con scp

Ejecutar la consola en VBGuest.

```
alumno@alumno-VirtualBox:~$
```

Copiar mediante scp VBGuest → BBB:

```
alu...:~$ scp ruta_archivo_en_VBGuest debian@192.168.100.13:/home/debian/ruta_en_BBB
```

Copiar mediante scp BBB → VBGuest:

```
alu...:~$ scp debian@192.168.100.13:/home/debian/ruta_archivo_en_BBB ruta_en_VBGuest
```

### 2.5.3 Comprobar puertos serie en BBB

Conectar a la BBB los dispositivos USB ↔ Serial que se van a utilizar en la aplicación. Utilizar un hub USB para más de un dispositivo.

Ejecutar en BBB para ver un listado de los dispositivos disponibles:

```
debian@beaglebone:~$ ls /dev/ttyU*  
/dev/ttyUSB0 /dev/ttyUSB1
```

Si hay varios, para distinguir quien es quien, remover de uno en uno y volver a ejecutar la instrucción anterior.

#### 2.5.4 Bridge para acceso a internet desde el SBC

Si se necesita que el SBC tenga acceso a Internet a través del VBGuest, se debe hacer mediante un script como el siguiente a ejecutar cada vez que se arranca una sesión en el VBGuest:

Each  
new  
session

```
alumno@alumno-VirtualBox:~$ sudo /opt/AlMar-Installs/set-BridgeForInet-sudo.sh
```

```
#!/bin/bash  
  
# Myself acting as bridge for Internet access from the SBC (https://medium.com/@TaranChinmai/sharing-internet-connection-from-a-linux-machine-over-ethernet-a5cbbd775a4f)  
IFACE_FOR_INTERNET=$(ip route get 8.8.8.8 | grep 8.8.8.8 | awk -v FS=' src' '{print $1}' | awk -v FS='dev ' '{print $2}')  
  
sysctl net.ipv4.ip_forward=1  
iptables -t nat -A POSTROUTING -o ${IFACE_FOR_INTERNET} -j MASQUERADE  
iptables -I FORWARD -o ${IFACE_FOR_INTERNET} -s 192.168.100.0/24 -j ACCEPT  
iptables -I INPUT -s 192.168.100.0/24 -j ACCEPT  
echo "AlMar INFO: Remember to run in target SBC: $ sudo route add default gw 192.168.100.100"
```

Algunas veces, el SBC que debe usar este puente no actualiza sus rutas, por lo que se debe hacer manualmente en el mismo (mediante sesión SSH):

```
debian@beaglebone:~$ sudo route add default gw 192.168.100.100
```

### 3 Uso de kit BBB en Qt Creator (programas modo consola)

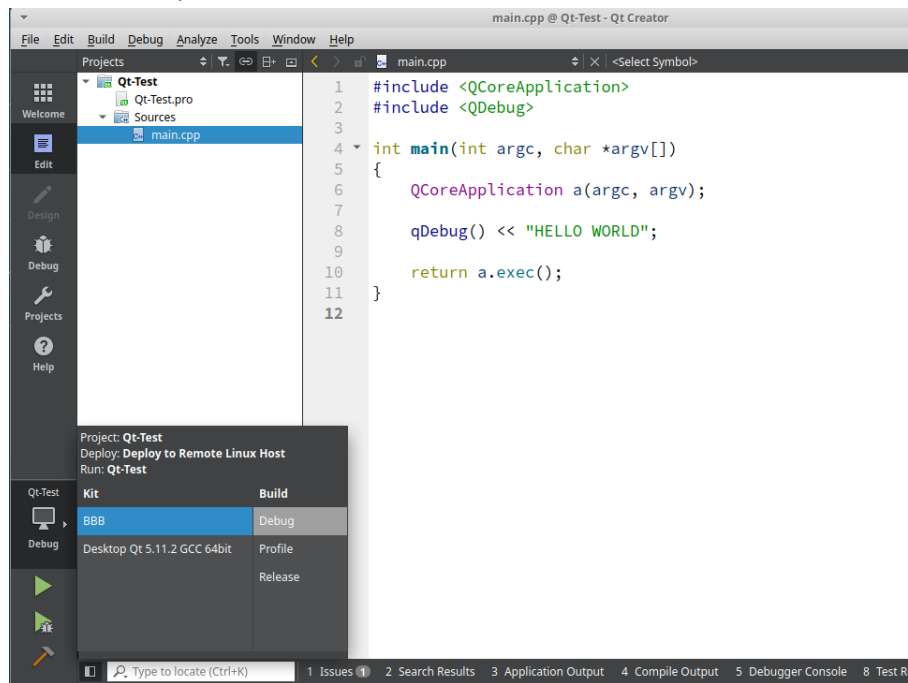
Cuando se crea un nuevo proyecto solicitará los kits para los que debe estar disponible: seleccionar ambos (Desktop y BBB-Debian).

#### 3.1 Con qmake

Añadir/modificar en archivo .pro

```
linux-beagle* {  
    ROOTFS = $$$(HOME)/Common-Installs/BBB/RootFS  
    INCLUDEPATH += $$${ROOTFS}/usr/include/arm-linux-gnueabi/hf  
}  
  
# Default rules for deployment.  
qnx: target.path = /tmp/${TARGET}/bin  
else: unix:!android: target.path = /home/debian/MyTests  
!isEmpty(target.path): INSTALLS += target
```

### 3.2 Seleccionar el kit a aplicar:



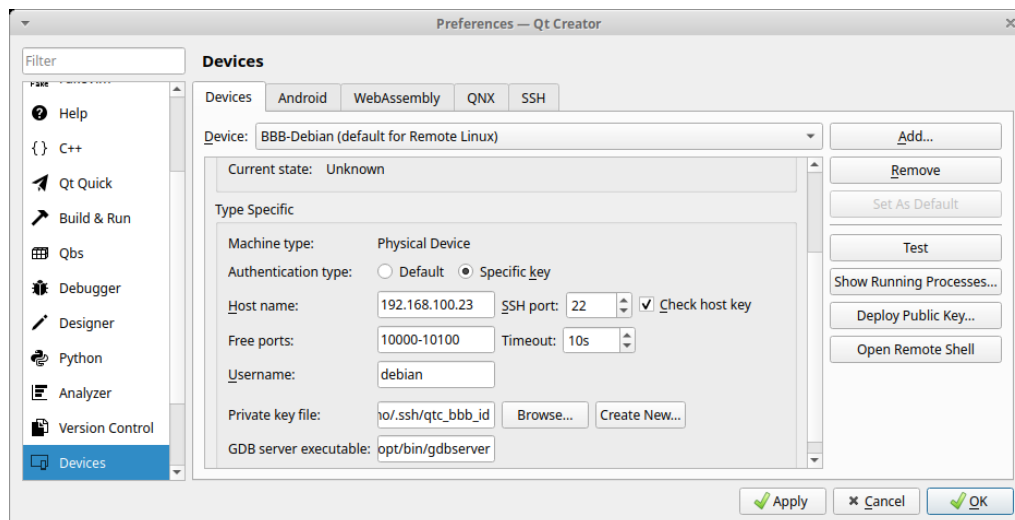
Si se selecciona el kit Desktop, la aplicación se ejecutará en el equipo de desarrollo.

Si se selecciona el Kit BBB-Debian, la aplicación se descargará y ejecutará en la BBB.

### 3.3 Modificar conexión de dispositivo (si es necesario)

En Qt-Creator, seleccionar Edit → Preferences → Devices, y ahí el dispositivo BBB-Debian

Las opciones que aparecerán son las siguientes:



Ejecutar Test para comprobar la conexión con la BBB. En caso de éxito, no hay más que hacer.



Si falla la conexión, hay que rehacer los archivos de claves criptográficas. Para ello, en la pantalla anterior de configuración:

Private key file. → [Create New ...]	
Private key file →: [Browse] Seleccionar archivo: /home/alumno/.ssh/qtc_bbb_id (Para navegar a este directorio, es necesario activar la opción Show Hidden Files con el botón derecho del ratón) [Save] [Generate And Save Key Pair]	
[Deploy Public Key] Seleccionar archivo: /home/alumno/.ssh/qtc_bbb_id.pub [Open] Passwd: tempPWD [Next] [Finish]	

### 3.4 Instalar aplicación para ejecución en el arranque de la BBB

Una vez se dispone de una aplicación operativa, se puede instalar como un servicio para que se ejecute en el arranque de la BBB.

El siguiente ejemplo supone que se desean ejecutar dos aplicaciones en el arranque:

```
/home/debian/MyTests/AplicacionAEjecutar1 param11 param12
```

```
/home/debian/MyTests/AplicacionAEjecutar2 param2
```

#### 3.4.1 Crear archivo con script de arranque

En el equipo host, con el editor de texto, crear archivo Shell con los ejecutables a arrancar (ej. initRover.sh):

```
#!/bin/bash
# ttyXXXX = 115200 baud (la app requiere que ttyXXXX tenga esta velocidad, sustituir
#                               XXXX por S1, USB0, USB1, etc.)
eval stty -F /dev/ttyXXXX 115200

# Arrancar programas
/home/debian/MyTests/AplicacionAEjecutar1 param11 param12 &
```



```
/home/debian/MyTests/AplicacionAEjecutar2 param2
```

### 3.4.2 Crear archivo de servicio

En el equipo host, con el editor de texto, crear archivo con el nombre del servicio (ej. initRover.service):

```
[Unit]
Description=init script
After=syslog.target network.target
[Service]
Type=simple
ExecStart=/home/debian/MyTests/initRover.sh
[Install]
WantedBy=multi-user.target
```

### 3.4.3 Copiar archivos a BBB

Se puede hacer desde ventana de comandos en el equipo host:

```
alu...:~$ cd $HOME/.../directorio_de_archivos_sh_y_service
alu...:~$ scp initRover.* debian@192.168.100.23:/home/debian/MyTests/
```

### 3.4.4 Instalar servicio en BBB

```
alu...:~$ ssh debian@192.168.100.23 (clave temppwd)
debian@beaglebone:~$ su (clave temppwd)
root@beaglebone:/home/debian# cp /home/debian/MyTests/*.service /etc/systemd/system/
root@beaglebone:/home/debian# systemctl start initRover.service
root@beaglebone:/home/debian# systemctl enable initRover.service
root@beaglebone:/home/debian# exit
debian@beaglebone:~$
```

En el siguiente arranque, la aplicación se ejecutará automáticamente.

### 3.4.5 Desinstalar servicio en BBB

```
$ ssh debian@192.168.100.23 (clave temppwd)
root@beaglebone:/home/debian$ su (clave temppwd)
root@beaglebone:/home/debian# systemctl stop initRover.service
root@beaglebone:/home/debian# systemctl disable initRover.service
root@beaglebone:/home/debian# exit
debian@beaglebone:~$
```

En el siguiente arranque, la aplicación no se ejecutará.