UNIVERSIDAD DE OVIEDO
Departamento de Ingeniería Eléctrica, Electrónica, de Computadores y Sistemas

Instalación de Arch-Linux en BeagleBone Black

Ignacio Alvarez García – Octubre 2018

Indice

Es necesario instalar la compilación cruzada desde Qt para BBB. No se puede utilizar el compilador tal cual porque genera código para PC (Intel-x64), mientras que la BBB tiene un microprocesador ARM-V7-hf.

Instalación de compiladores para ARM-V7-hf

- Entrar en /home/developer/BBB-Installs (o directorio similar que se haya creado):

```
$ cd /home/developer/BBB-Installs
```

- Descargar y descomprimir compilador para ARM:

```
$ wget -c https://releases.linaro.org/components/toolchain/binaries/6.4-2018.05/arm-linux-gnueabihf/gcc-linaro-6.4.1-2018.05-x86_64_arm-linux-gnueabihf.tar.xz
$ tar -xvf gcc-linaro-6.4.1-2018.05-x86_64_arm-linux-gnueabihf.tar.xz
$ export CC=`pwd`/gcc-linaro-6.4.1-2018.05-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-
```

Instalación de Qt5 para ARM-V7-hf

- Crear directorio RootFS y descomprimir el root file system que ya se había instalado en la SD/EMMC anteriormente:

```
$ mkdir RootFS
$ sudo bsdtar -xpf ArchLinuxARM-am33x-latest.tar.gz -C RootFS
```

- Cambiar enlaces simbólicos en directorios RootFS:

```
$ wget https://raw.githubusercontent.com/riscv/riscv-poky/priv-1.10/scripts/sysroot-relativelinks.py
$ chmod +x sysroot-relativelinks.py
$ sudo ./sysroot-relativelinks.py RootFS
```

- Descargar última versión de Qt5 desde git:

```
$ mkdir -p git
$ cd git
$ git clone git://code.qt.io/qt/qt5.git
$ cd qt5
$ perl init-repository --module-subset=qtbase,qtserialport,qtimageformats
```

- Configurar y compilar:

```
$ ./configure -v -qpa linuxfb -no-opengl -device linux-rasp-pi-g++ -device-option CROSS_COMPILE=$CC -sysroot /home/developer/BBB-Installs/RootFS -opensource -confirm-license  -optimized-qmake -reduce-exports -release -make libs -nomake examples -qt-zlib -qt-libpng -qt-libjpeg -prefix /opt/Qt/BBB

$ make –j4
$ sudo make install
```

Instalación de depurador remoto

- Descargar fuentes de gdb y compilar para ARM-V7-hf:

```
$ cd /home/developer/BBB-Installs
$ wget http://ftp.gnu.org/gnu/gdb/gdb-8.2.tar.gz
$ tar -xvf gdb-8.2.tar.gz
$ cd gdb-8.2/gdb/gdbserver
$ ./configure --host=arm-linux-gnueabihf --prefix=/home/developer/BBB-Installs/Roo
tFS/opt CC=${CC}gcc CXX=${CC}g++ CXXLD=${CC}g++
$ make -j4
$ sudo make install
```

- Instalar gdb-multiarch:

```
$ sudo apt-get install gdb-multiarch
```

Copia de archivos instalados en ARM

Es necesario copiar los archivos en /opt, que sólo está permitido para el usuario root, así que lo primero es desbloquear el acceso ssh como root:

Entrar en BBB con ssh:

```
$ ssh alarm@192.168.100.23
$ su   (clave root)
# nano /etc/ssh/sshd_config
```

Cambiar la línea:

```
#PermitRootLogin prohibit-password
```

Por:

```
PermitRootLogin yes
```

En la sección #Ciphers, añadir línea:

```
Ciphers +aes128-cbc
```

Guardar archivo.

Reiniciar servicio sshd y volver a usuario alarm:

```
# systemctl restart sshd
# exit
$
```

Abrir dos ventanas del gestor de archivos, una con la carpeta `/home/developer/BBB-Installs/RootFS/opt` y otra con la carpeta `sftp://root@192.168.100.23/opt/` (clave root)

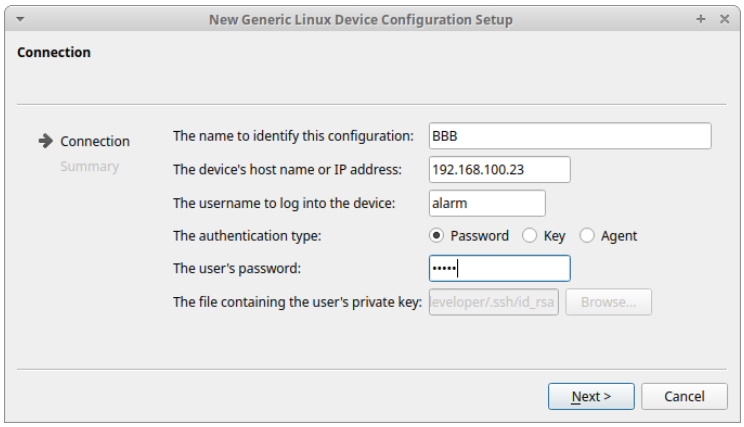
Copiar archivos de la 1ª carpeta a la 2ª.

Una vez finalizada la copia, cerrar ambas carpetas, y restaurar si se desea el PermitRootLogin al estado inicial (permitir login de root con ssh se considera un agujero de seguridad).

Instalación de kit en Qt Creator

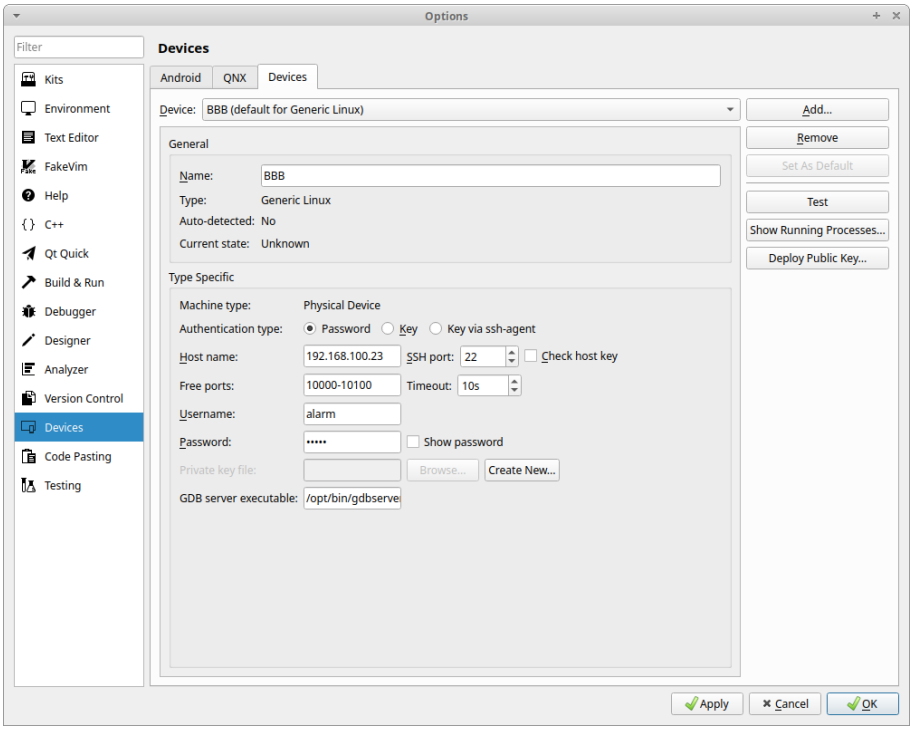
- Crear nuevo dispositivo

  Arrancar QtCreator, e ir a Tools -> Options
  * Devices -> Add... Generic Linux Device
  Name : BBB
  IP : 192.168.100.23
  User : alarm
  Pass: alarm
  [Next]



  Una vez añadido, escribir el nombre del depurador remoto:

  GDB server executable: /opt/bin/gdbserver

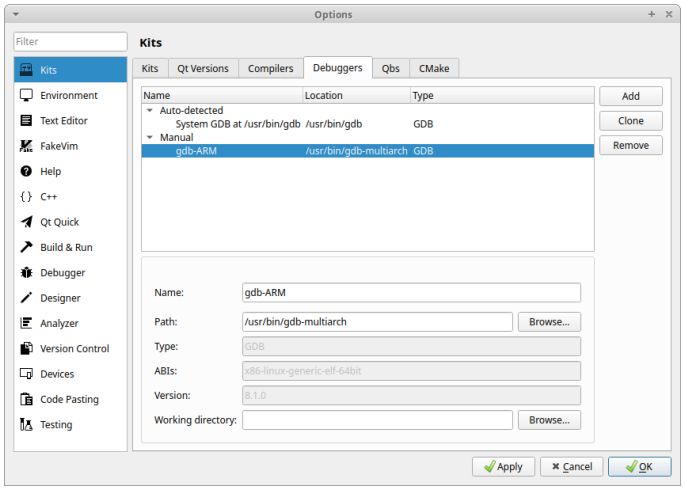


Pulsar [Apply]

- Añadir depurador local:

  Dentro de Options, seleccionar Kits -> Debugger y añadir nuevo depurador:

  - Name: gdb-ARM
  - Path: /usr/bin/gdb-multiarch



Pulsar [Apply]

- Añadir compiladores:

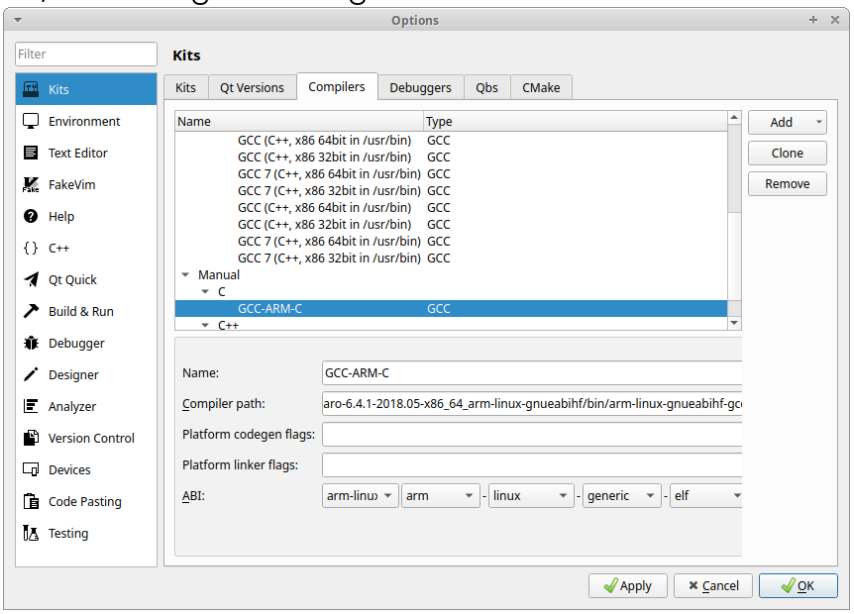
  Dentro de Options, seleccionar Kits -> Compiler y añadir nuevos compiladores GCC para C++:

  - Name: gcc-ARM-C++

- Path: /home/developer/BBB-Installs/gcc-linaro-6.4.1-2018.05-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-gcc

Y para GCC C:

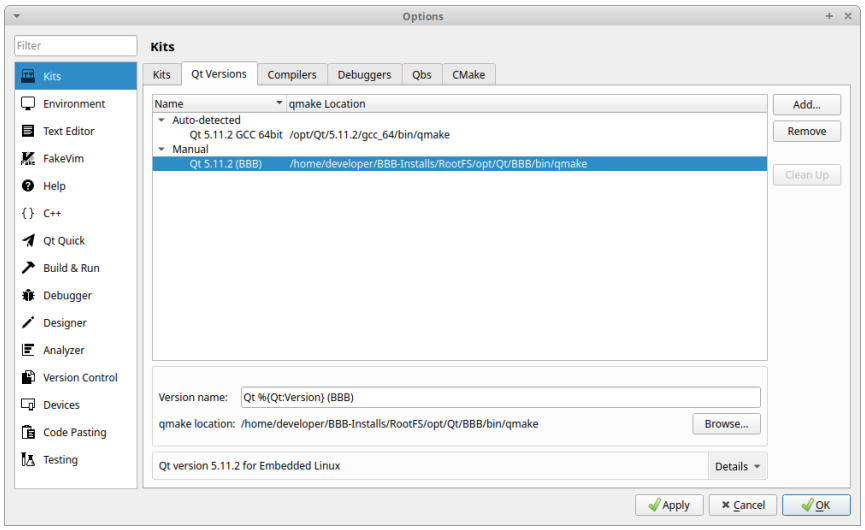
- Name: gcc-ARM-C
- Path: /home/developer/BBB-Installs/gcc-linaro-6.4.1-2018.05-x86_64_arm-linux-gnueabihf/bin/arm-linux-gnueabihf-gcc



Pulsar [Apply]

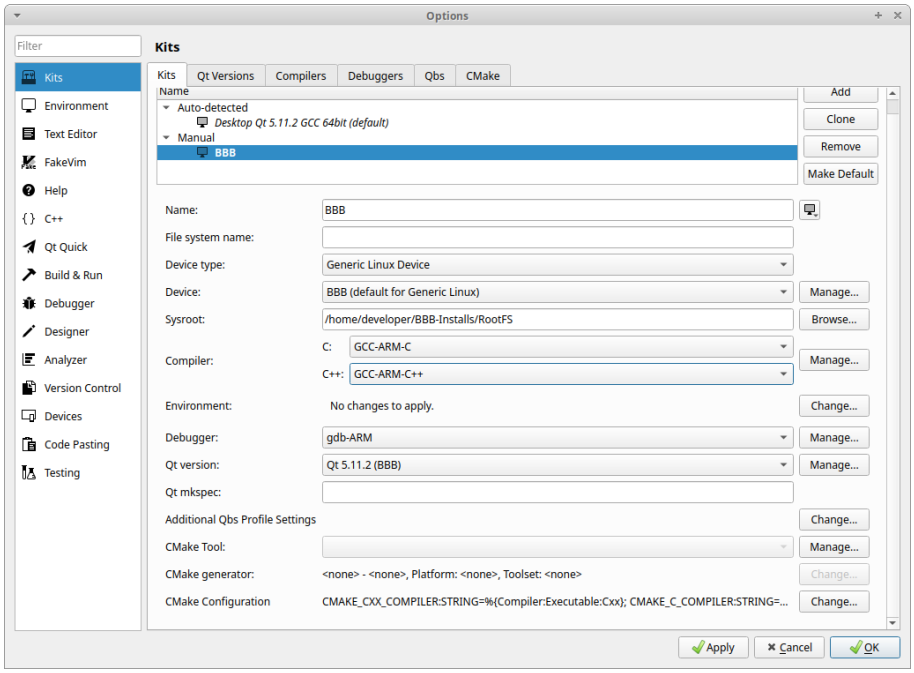
- Añadir versión Qt:
  Dentro de Options, seleccionar Kits -> Qt Versions y añadir versión de Qt seleccionando el path para qmake:

  /home/developer/BBB-Installs/RootFS/opt/Qt/BBB/bin/qmake



Pulsar [Apply]

- Finalmente, añadir kit BBB, utilizando los elementos anteriores:
  - Dentro de Options, seleccionar Kits -> Kits y añadir nuevo kit:
    Name : BBB
    Device type : Generic Linux Device
    Device : BBB
    Sysroot: /home/developer/BBB-Installs/RootFS
    Compiler C: GCC-ARM-C
    Compiler C++: GCC-ARM-C++
    Debugger: gdb-ARM
    Qt versión: Qt-5.11.2 (BBB)



Pulsar [Apply]

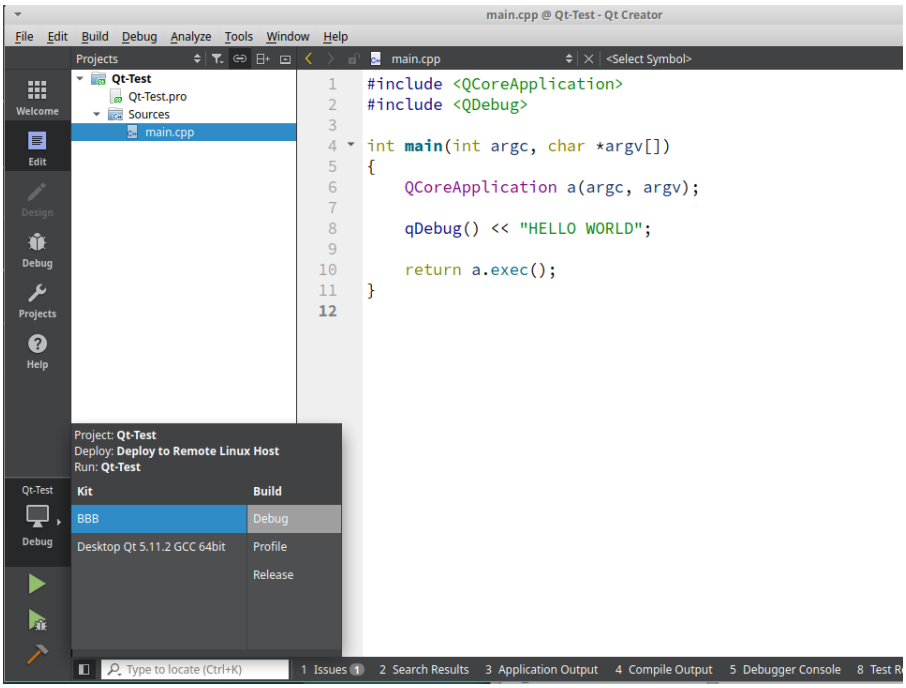
Uso de kit en Qt Creator (programas modo consola)

Una vez instalado el nuevo kit, cuando se crea un nuevo proyecto solicitará los kits para los que debe estar disponible: seleccionar ambos (Desktop y BBB).

Modificar archivo .pro, cambiando el directorio para el deployment (target.path), que debe existir en la BBB (si no existe, crearlo desde BBB con `$ mkdir /home/alarm/Tests`):

```
# Default rules for deployment.
qnx: target.path = /tmp/$${TARGET}/bin
else: unix:!android: target.path = /home/alarm/Tests/
!isEmpty(target.path): INSTALLS += target
```

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, la aplicación se descargará y ejecutará en la BBB.

Uso de kit en Qt Creator (programas con widgets)

También se pueden ejecutar en la BBB programas tipo gráfico (con widgets). En este caso, es necesario:

- Copiar en la carpeta adecuada del disco de la BBB (micro-SD o EMMC) los archivos con fuentes (tipos de letra):

```
$ mkdir /home/developer/BBB-Installs/fonts
$ find /usr/share/fonts -name *.ttf -exec cp {} /home/developer/BBB-Installs/fonts/ \;
```

  Mediante una ventana de navegador de archivos, entrar como root en sftp://192.168.100.23, y moverse a la carpeta /opt/Qt/BBB/lib. Crear en ella el directorio fonts y moverse a dicho directorio.

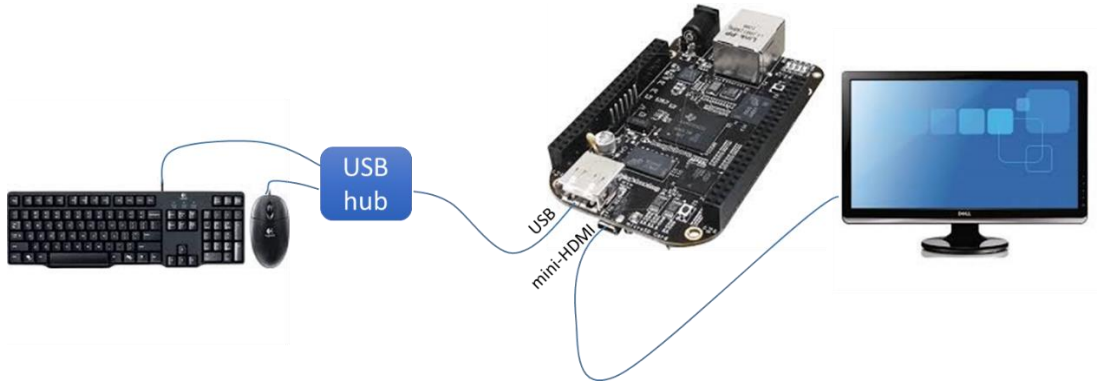
  En otra ventana del navegador, abrir la carpeta /home/developer/BBB-Installs/fonts.

  Copiar los contenidos de la 2ª carpeta a la 1ª.

- Autorizar al usuario 'alarm' a utilizar la pantalla de vídeo:

```
$ ssh alarm@192.168.100.23     (clave alarm)
[alarm@alarm ~]$ su            (clave root)
[root@alarm alarm]# usermod -a -G video alarm
[root@alarm alarm]# reboot now       (provocará un reinicio de la BBB)
```

- Ya se puede conectar una pantalla al conector mini-HDMI disponible, y teclado/ratón a través del USB, y ejecutar aplicaciones con widgets:



Instalar aplicación para ejecución en el arranque

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/alarm/Tests/AplicacionAEjecutar1 param11 param12

/home/alarm/Tests/AplicacionAEjecutar2 param2

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
# ttyS1 = 115200 baud
eval stty -F /dev/ttyS1 115200

# Arrancar programas
/home/alarm/Tests/AplicacionAEjecutar1 param11 param12 &
/home/alarm/Tests/AplicacionAEjecutar2 param2
```

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/alarm/Tests/initRover.sh
[Install]
WantedBy=multi-user.target
```

Copiar archivos a BBB

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

```
$ cd /home/developer/…./directorio_de_archivos_sh_y_service
$ scp initRover.* alarm@192.168.100.23:/home/alarm/Tests/
```

Instalar servicio en BBB

```
$ ssh alarm@192.168.100.23     (clave alarm)
[alarm@alarm ~]$ su            (clave root)
[root@alarm alarm]# cp /home/alarms/Tests/initRover.service /etc/systemd/system/
[root@alarm alarm]# systemctl start initRover.service
[root@alarm alarm]# systemctl enable initRover.service
```

En el siguiente arranque, la aplicación se ejecutará automáticamente.

Desinstalar servicio en BBB

```
$ ssh alarm@192.168.100.23     (clave alarm)
[alarm@alarm ~]$ su            (clave root)
[root@alarm alarm]# systemctl stop initRover.service
[root@alarm alarm]# systemctl disable initRover.service
```

En el siguiente arranque, la aplicación no se ejecutará.