Dentro del Eclipse (Con la BBB ya conectada):

----------PASO PREVIO (Tener instaladas y configuradas todas las cosas para poder hacer compilación cruzada (es decir para ARM, y si tenemos el emulador Qemu también podremos iniciar esos programas compilados en el PC).

La parte de instalar el Qemu en el PC nos la podemos saltar ya que no tiene mucha utilidad ejecutar código pensado para la BBB en el PC.

Si da algún error, habrá que bajar otra versiones de los paquetes que ponga en el libro.

Si da algún error del linker o del compilador, habrá que hacer algo así con las librerias en las que nos salga error, sustituyendo las rutas por las librerías que nos den error:

cd /

sudo find ./ | grep libpthread.so

cp ./arm-linux-gnueabihf/lib/libpthread.so /usr/lib/

cp ./arm-linux-gnueabihf/lib/libpthread.so.0 /usr/lib/

EXPLICADO desde la página 251 hasta la 261 del libro Exploring BeagleBone.

----------CREAR PROYECTO (o importar):

File ➢ New ➢ C++ project. As illustrated in

Figure 7-1 (a), set the project name to “BBBTest,” pick the project type “Hello

World C++ Project,” and the Toolchain to be “Cross GCC.” Then click Finish.

Repeatedly click Next until you see the “Cross GCC Command,” as illustrated

in Figure 7-1 (b). Enter arm‐linux‐gnueabihf‐ for the cross-compiler prefi x and

set the path to /usr/bin . Finally, click Finish.

Project ➢ Properties ➢ C/C++->

General ➢ Paths and Symbols, and set the following (according to your ver-

sion of gcc and g++):

Includes (tab) ➢ GNU C (Include directories) ➢ Add ➢ File System ➢

File System ➢ /usr/arm‐linux‐gnueabihf/include

Includes (tab) ➢ GNU C++ (Include directories) ➢ Add ➢ File System ➢

File System ➢ /usr/arm‐linux‐gnueabihf/include/c++/4.7.2 or your

current version.

Library Paths (not Libraries) ➢ Add ➢ File System ➢ File System ➢ /usr/

arm‐linux‐gnueabihf/lib

Click OK to apply the confi guration

--------CONECTARSE A LA BBB DESDE ECLIPSE:

Go to Window ➢Show View ➢ Other ➢ Remote Systems ➢ Remote Systems.

In the Remote Systems frame that appears, click the icon for Define a Connection to a Remote System, and in the New Connection dialog, select the following:

■ Choose Linux Type ➢ Next.

■ Host Name: Enter your BBB IP address—e.g., 192.168.7.2

■ Connection Name: Change it to “BeagleBone Black” ➢ Next

■ [Files] Confi guration ➢ ssh.fi les ➢ Next

■ [Processes] Configuration ➢ processes.shell.linux ➢ Next

■ [Shells] Configuration ➢ ssh.shells ➢ Next

■ [Ssh Terminals] (no change) ➢ Finish

You can then right‐click BeagleBone Black in the bottom frame and choose

Connect.

Luego botón derecho en ssh terminals y launch terminal dentro de la conexión de la BBB para ejecutar la consola de la BBB dentro del Eclipse. Recuerda tener cerradas todas las otras sesiones de la BBB en las otras terminales.

------EJECUTAR PROGRAMA COMPILADO PARA LA BBB:

Desde la terminal se ejecuta el archivo (nombreProyecto) que será el ejecutable obtenido de la compilación, y se encuentra en la carpeta (nombreProyecto)/Debug/

Si se le da al play desde el Eclipse, se ejecuta en el PC, no en la BBB.

También se le puede dar al botón de debug como se explica en la siguiente sección, y no poner ningún breakpoint, así sería como ejecutarlo normal.

------DEBUGGEAR PROGRAMA COMPILADO PARA LA BBB:

\*--Paso previo (Haber configurado el gdb en la BBB y el PC).\* (NO explicado aqui)

(Explicado en página 266 del libro Exploring BeagleBone)

On the desktop machine, create a file called .gdbinit in the project folder

using nano and add the line:

set architecture arm

(Si hay problema de permisos, crearlo directamente en la raíz del workspace)

Go to Run ➢Debug Configurations ➢ Debugger, and delete any current debug confi gura-

tions. Select C/C++ Remote Applications on the left‐hand side and right‐click

it to create a new configuration. In this example it is called “BBBTest Debug”

Change the debugger command from gdb to **/usr/bin/gdb-multiarch** , as illustrated

in Figure 7-8 . You also need to attach the .gdbinit file that was just created.

Click the Browse button to the right of “GDB command File” and locate your

workspace directory. You may have to right‐click the File Explorer window and

choose Show hidden files to find the file .gdbinit . That confi guration fi le can

be used to set many more configuration options. For example, it can be used to

confi gure the remote server and choose different breakpoints.

Any program arguments can be added to the Arguments tab.

The argument is passed to the application that is executing on the BBB when the gdbserver command executes.

Finally, under the Gdbserver Settings tab, set the BBB’s IP (192.168.7.2), the executable

path and an arbitrary port number(el 12345 en este ejemplo) for the gdbserver command. This allows the desktop computer to remotely invoke the gdbserver command on the BBB

and to connect to it over TCP using its port number.

**Ahora CADA VEZ QUE LO QUERAMOS DEPURAR:**

NOTA: Hay que hacer build desde el eclipse(aunque falle) para que nos deje darle al botón de debug, se le pone el nombre del ejecutable (medcape) en la configuración del debug.

En la BBB (desde la ruta raíz de la BBB) ejecutar el siguiente comando, sustituyendo la ruta por la que sea en la que esté nuestro proyecto:

**cd**

**gdbserver ubuntu:12345 workspace2016/test2/Debug/test2**

Y desde el Eclipse lanzamos el debug que acabamos de configurar.

**------------EXTRA: Integración con Github desde Eclipse-**

Help ➢ Install New Software and choose “Luna . . .” (o el Eclipse que tu tengas) in the Work

section.

Ahora add->e introduces: http://download.eclipse.org/egit/updates

Select all (menos el de incubation)->next.

Luego desmarcar:

-Command Line Interface for Java implementation of Git 4.5.0.201609210915-r

-Java implementation of Git - optional LFS support 4.5.0.201609210915-r

Once this plug‐in is installed, you can open Window ➢ Show View ➢ Other ➢

Git, and choose Git Repositories, you then get a dialog with the option to “Clone a Git repository” ➢ GitHub, and you can search for <https://github.com/jrecaspiorno/medcape.git>