

HomeBrain Argentina

Hardware y Software basado en IoT para implementar Industry 4.0

Cómo configurar fácilmente Eclipse Neon 4.6 para realizar Cross-Compilation en módulos con procesador i.MX6UL

How to Easily Configure Eclipse Neon 4.6 to Perform Cross-Compilation on Modules with i.MX6UL Processor

Este documento se ha creado con el objetivo de ayudar a configurar la interfaz de desarrollo Eclipse Neon dentro del sistema operativo Ubuntu V16.04.3 LTS y así poder realizar Cross-Compilation de aplicaciones en C/C++ para ejecutarlas en forma remota directamente en el kit de desarrollo **Digi ConnectCore i.MX6UL Starter Kit** (en adelante **ccimx6ulstarter**).

El módulo **ccimx6ulstarter** deberá estar conectado a la misma red a la que se encuentre conectada la PC de desarrollo, así se podrá acceder a través de SSH al mismo. Para esto es recomendable que configure previamente la conexión WiFi del mismo así el módulo podrá tener acceso tanto a internet como a su PC para poder continuar.

HomeBrain utiliza sus propias imágenes, al momento de realizar este documento se utilizaba la versión **Digi Yocto V2.2-r2.2 (Morty)** precompilada, es por esto que se verán en el documento estas versiones de imagen de sistema operativo y SDK.

Asimismo la sintaxis y ejemplos sirven para todas las versiones previas **Digi Yocto V2.2-r#.#**; solo debe cambiarse el archivo sobre el cual se deba ejercer alguna acción.

Estamos trabajando actualmente en la generación de un documento completo en donde se indique como generar sus propias imágenes de Yocto así como sus correspondientes SDK.

Índice:

| | |
|--------------------------------------------------------------------------------------|----|
| 1) Antes de comenzar. | 3 |
| 2) Pasos para instalar y dejar ejecutable Eclipse Neon 4.6. | 4 |
| 3) Configuración de plug-ins de Eclipse Neon. | 7 |
| 4) Configurar plug-in Yocto Project SDK. | 9 |
| 5) Creación de una aplicación “Hola Mundo” para terminar de configurar Eclipse Neon. | 10 |
| 6) Creación de una conexión SSH entre Eclipse Neon y el módulo ccimx6ulstarter. | 13 |
| 7) Configuración del debugger. | 15 |
| 8) Configurar Run. | 17 |
| 9) Notas finales. | 20 |

1) Antes de comenzar:

Las siguientes tareas deberán estar realizadas antes de comenzar:

- Tener instalado al menos Digi Yocto V2.2-r1 (Morty) previamente en la placa ccimx6ulstarter: <https://www.digi.com/resources/documentation/digidocs/90001548/>
- Chequear que puede conectarse mediante SSH a la placa ccimx6ulstarter con el objetivo de que el IDE Eclipse Neon y descargar los archivos compilados en forma directa.
- Instalar el **SDK de Digi Yocto V2.2-r1 (Morty)** o superior en la PC de desarrollo en el directorio **“/opt”**.

<https://www.digi.com/resources/documentation/digidocs/90001548/>

```
user@ubuntu:/$ cd /opt
user@ubuntu:/opt$ sudo mkdir dey-2.2-r2.2_SDK
user@ubuntu:/opt$ sudo chown -R user:user /opt/dey-2.2-r2.2_SDK
user@ubuntu:/opt$ cd dey-2.2-r2.2_SDK
user@ubuntu:/opt/dey-2.2-r2.2_SDK$ ls -la
total 8
drwxr-xr-x 2 user user 4096 ago 21 11:42 .
drwxr-xr-x 7 root root 4096 ago 21 11:42 ..
```

```
user@ubuntu:/opt/dey-2.2-r2.2_SDK$ /home/user/Downloads/./dey-glibc-x86_64-core-image-
base-cortexa7hf-neon-toolchain-2.2-r2.sh
```

Digi Embedded Yocto SDK installer version 2.2-r2

=====

Enter target directory for SDK (default: /opt/dey/2.2-r2): **/opt/dey-2.2-r2.2_SDK**

A medida que el script de shell bash lo pida, ir completando las diferentes solicitudes y finalizar.

- En una estación de desarrollo ejecutando Ubuntu 16.04.3 LTS o superior instalar los siguientes paquetes:

```
user@ubuntu:/$ sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib build-
essential chrpath socat libstdc++12-dev xterm make xsltproc docbook-utils fop dblatex xmlto
python-git
```

- Contar con la última versión de Oracle Java 1.8 instalada:

```
user@ubuntu:/$ sudo add-apt-repository ppa:webupd8team/java
user@ubuntu:/$ sudo apt-get update
user@ubuntu:/$ sudo apt-get install oracle-java8-installer
user@ubuntu:/$ sudo apt-get install oracle-java8-set-default
user@ubuntu:/$ java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
```

Con el último comando verificamos si realmente se ha instalado correctamente la versión 144 (en nuestro caso) de Oracle Java 1.8.

2) Pasos para instalar y dejar ejecutable Eclipse Neon 4.6:

- a) Descargar Eclipse Neon 4.6 de la página oficial del IDE: <http://www.eclipse.org/neon/>

En nuestro caso el directorio de descarga de Eclipse Neon ha sido “/home/user/Downloads”.

- b) Crear directorio de instalación:

```
user@ubuntu:/$ cd /opt
user@ubuntu:/opt$ sudo mkdir eclipse-neon
user@ubuntu:/opt$ sudo chown -R user:user /opt/eclipse-neon
user@ubuntu:/opt$ cd eclipse-neon
user@ubuntu:/opt/eclipse-neon$ ls -la
total 8
drwxr-xr-x 2 user user 4096 ago 21 11:42 .
drwxr-xr-x 7 root root 4096 ago 21 11:42 ..
```

Por favor verificar que los permisos y propietarios de la carpeta “/opt/eclipse-neon” hayan cambiado al usuario “user” (o su nombre de usuario), caso contrario Eclipse Neon no podrá tener acceso al directorio y no podrá ejecutarse.

- c) Descomprimos Eclipse Neon en el directorio “/opt/eclipse-neon”:

```
user@ubuntu:/opt/eclipse-neon$ tar xvf /home/user/eclipse-cpp-neon-3-linux-gtk-
x86_64.tar.gz
user@ubuntu:/opt/eclipse-neon$ ls -la
total 496
drwxr-xr-x 8 user user 4096 ago 21 10:35 .
drwxr-xr-x 6 root root 4096 ago 21 13:02 ..
-rw-r--r-- 1 user user 179491 ago 20 20:48 artifacts.xml
drwxr-xr-x 11 user user 4096 ago 21 10:35 configuration
drwxr-xr-x 2 user user 4096 mar 14 14:26 dropins
-rwxr-xr-x 1 user user 80385 mar 01 08:00 eclipse
-rw-r--r-- 1 user user 469 ago 20 20:48 eclipse.ini
drwxr-xr-x 101 user user 12288 ago 20 20:48 features
-rwxr-xr-x 1 user user 140566 mar 01 08:00 icon.xpm
-rw-rw-r-- 1 user user 6297 feb 11 2017 notice.html
drwxr-xr-x 5 user user 4096 ago 21 10:35 p2
drwxr-xr-x 11 user user 49152 ago 20 20:48 plugins
drwxr-xr-x 2 user user 4096 mar 14 14:26 readme
```

Verificamos de esta manera que todos los archivos tengan como propietario al usuario “user”. Por otro lado, veremos que el archivo binario “eclipse” se encuentra dentro del directorio, el mismo servirá para ejecutar la interfaz de desarrollo.

- d) Creamos un acceso directo para poder facilitar la ejecución de Eclipse Neon:

```
user@ubuntu:/$ cd /home/user/.local/applications/share
user@ubuntu:/$ gedit eclipseneon.desktop
```

- En el editor de textos Gedit o similar ingresar las siguientes líneas:

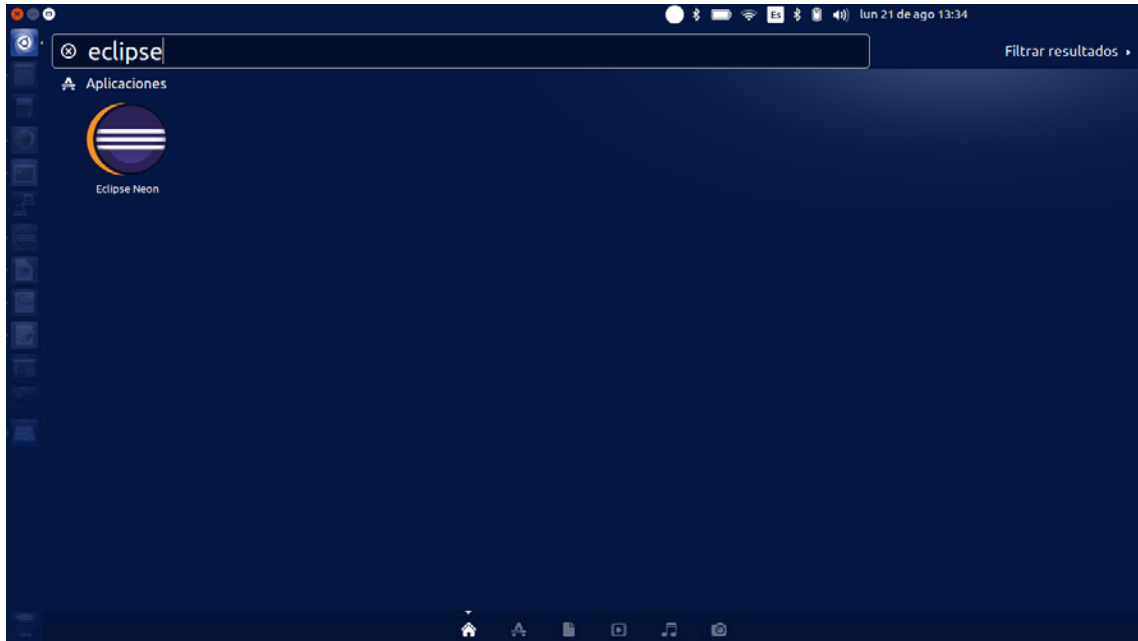
```
[Desktop Entry]
Version=4.6
Name=Eclipse Neon 4.6
Type=Application
Exec=/opt/eclipse-neon/eclipse
```

Icon=/opt/eclipse-neon/icon.xpm

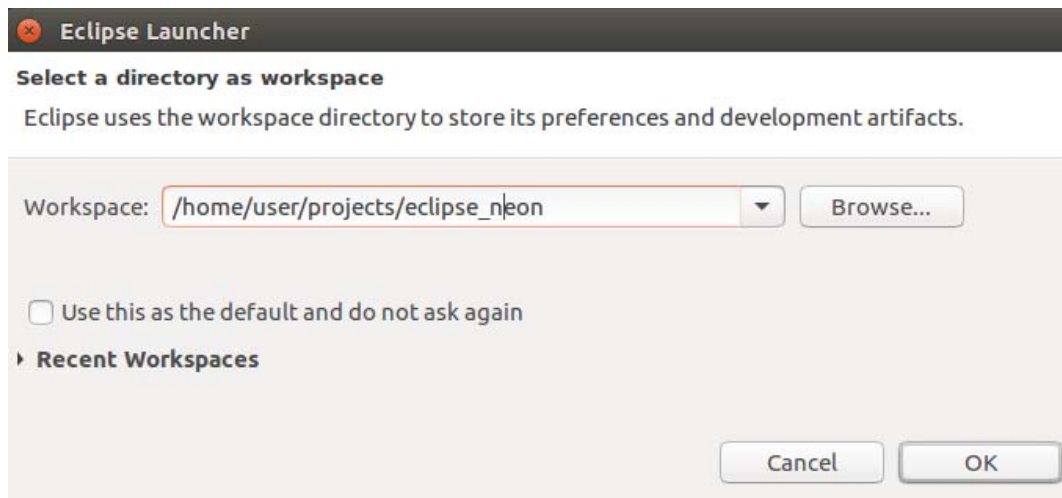
Terminal=false

Comment=Eclipse IDE 4.6

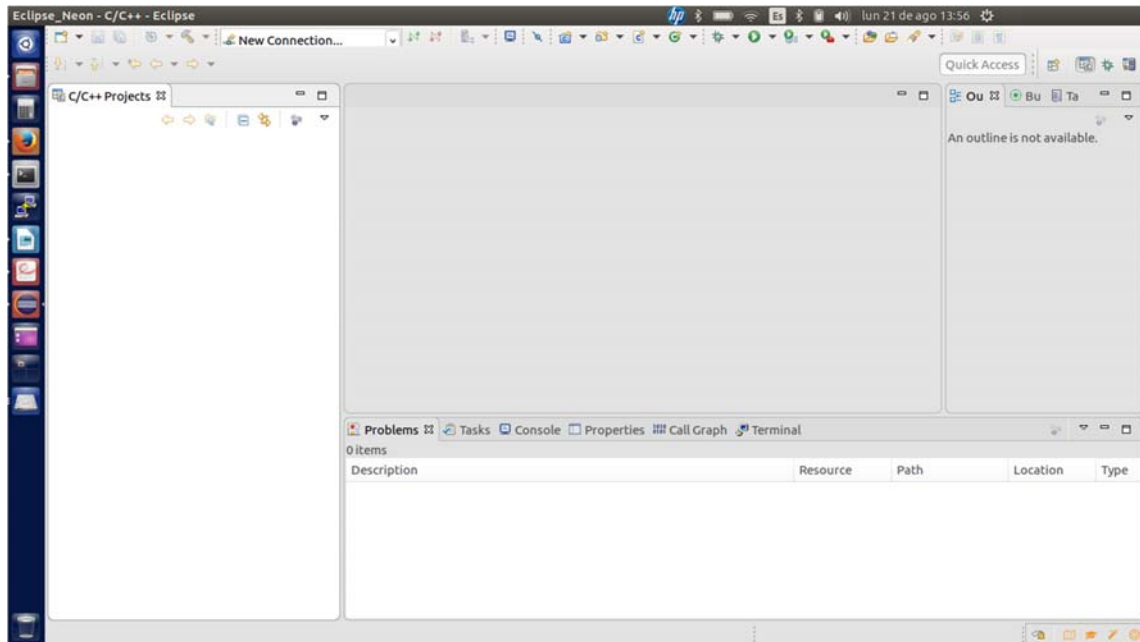
- Guardar el archivo y cerrarlo.
- Luego de esto buscaremos en las aplicaciones de Unity (primer ícono en el panel lateral de Ubuntu):



- Arrastre el ícono de Eclipse Neon a la barra lateral para fijarlo. Así quedará el ícono disponible para lanzar el programa.
- Ejecute Eclipse Neon:



- Eclipse Neon ejecutado tendrá esta apariencia:



3) Configuración de plug-ins de Eclipse Neon:

a) Ingresar a la opción del menú **Help**→**Install New Software...**

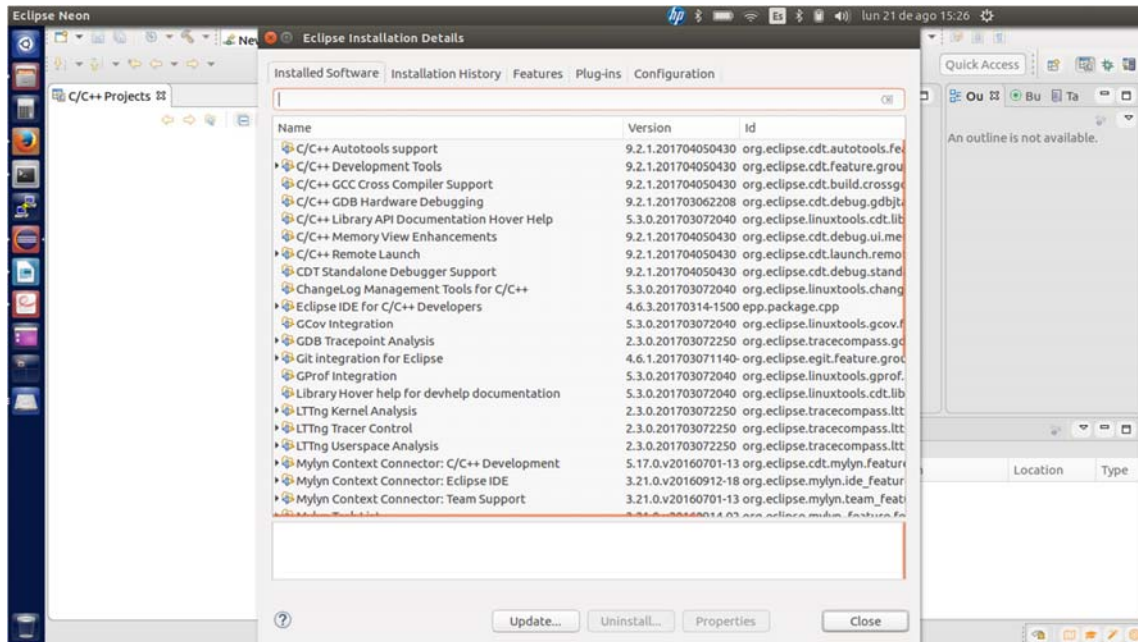
b) Instalaremos las herramientas pertenecientes a Eclipse Neon:

- Para esto ingresar en el textbox **“Working with”** el siguiente link:
<http://download.eclipse.org/releases/neon>
- Instalar las siguientes herramientas:
 - **Linux Tools:**
 - ✓ **LTTng – Linux Tracing Toolkit** (si no figurara entre las opciones es que ya se encuentra instalada).
 - **Mobile and device development:**
 - ✓ **C/C++ Remote Launch.**
 - ✓ **Remote System Explorer User Actions.**
 - ✓ **TM Terminal.**
 - ✓ **TCF Remote System Explorer add-in.**
 - ✓ **TCF Target Explorer.**
 - **Programming Languages:**
 - ✓ **C/C++ Autotools Support.**
 - ✓ **C/C++ Development Tools.**

c) Instalamos el **plug-in** de **Yocto Morty V2.2.1**:

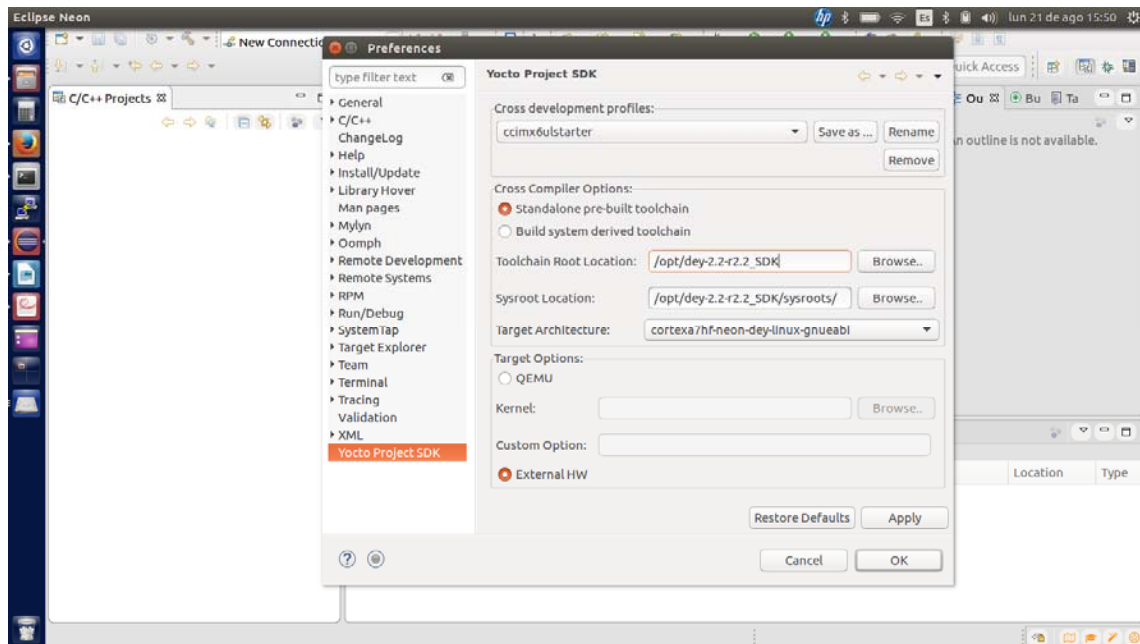
- Para esto ingresar en el textbox **“Working with”** el siguiente link:
<https://downloads.yoctoproject.org/releases/eclipse-plugin/2.2.1/neon/>
- Instalar las siguientes herramientas:
 - **Yocto Project Documentation plug-in.**
 - **Yocto Project SDK Plug-in.**
- Una vez instaladas las herramientas estará en condiciones de configurar Eclipse Neon para poder realizar Cross-Compilation.

- Puede verificar que todas las herramientas han sido instaladas en forma satisfactoria ingresando a **Help→About Eclipse** y presionar el botón **Installation details**:



4) Configurar plug-in Yocto Project SDK:

- Ingresar a **Windows** → **Preferences** → **Yocto Project SDK**:

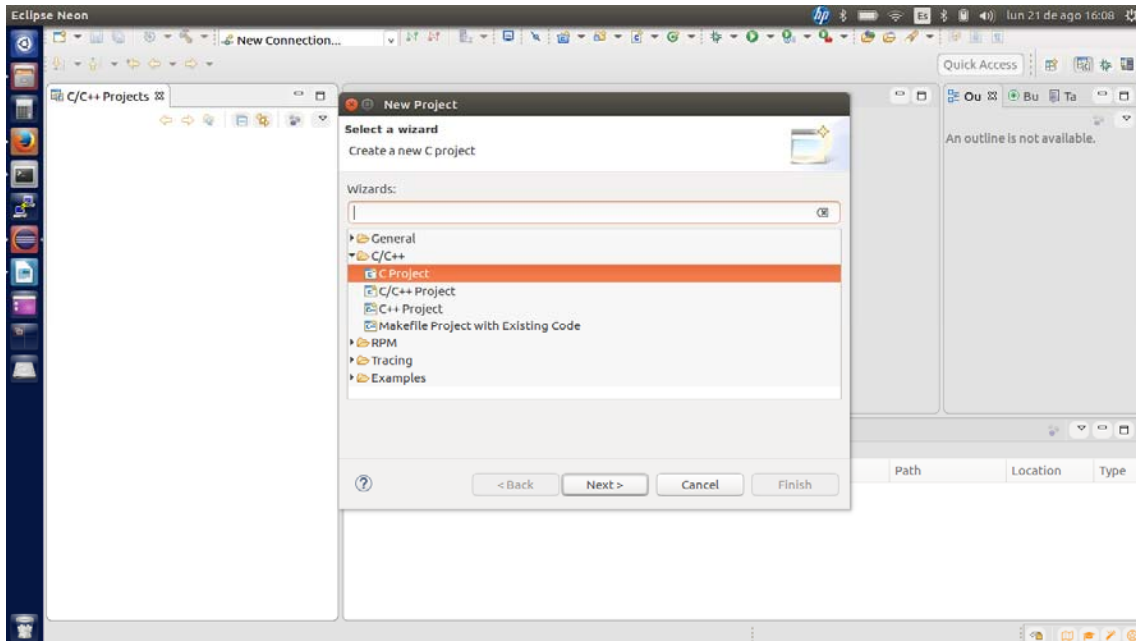


- Seleccionar en la opción de **“Cross Compiler Options”**: **Standalone pre built toolchain**.
- Ingresar en el textbox **“Toolchain Root Location”** el directorio **/opt/dey-2.2-r2.2_SDK**.
- Ingresar en el textbox **“Sysroot Location”** el directorio **/opt/dey-2.2-r2.2_SDK/sysroots**.
- En **“Target Architecture”** aparecerá directamente la arquitectura que tiene la plataforma **ccimx6ulstarter**.
- En **“Target Options”** seleccionar: **External HW**. Esta opción se utiliza cuando se tiene conectada directamente la placa de desarrollo ccimx6ulstarter a los puertos USB directamente.
- Presionar el botón **Apply** y luego **OK**.

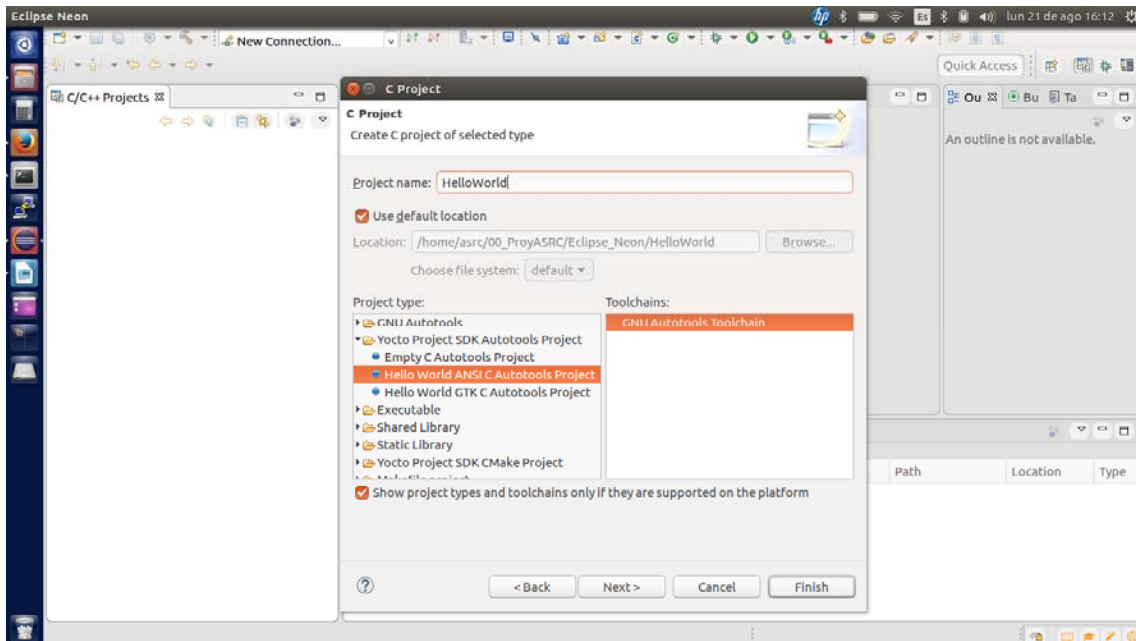
Estos pasos son fundamentales, si estuviesen mal configurados, al querer compilar, dará error debido a que no encuentra las fuentes para realizar el Cross-Compilation.

5) Creación de una aplicación “Hola Mundo” para terminar de configurar Eclipse Neon:

- Crear proyecto presionando **File→New→Project...**:

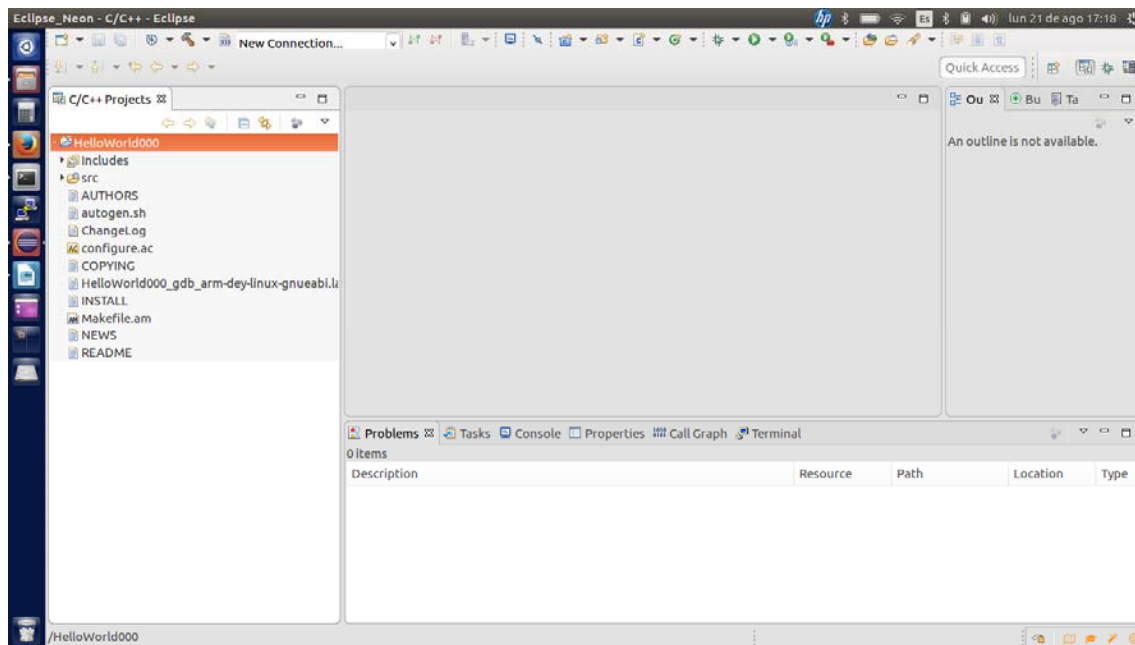


- Seleccionar “Yocto Project SDK Autotools Project” y “Hello World ANSI C Autotools Project”.
- Ingresar el nombre del proyecto, preferentemente sin caracteres especiales como ser guiones o signos de puntuación para que el compilador no arroje errores.

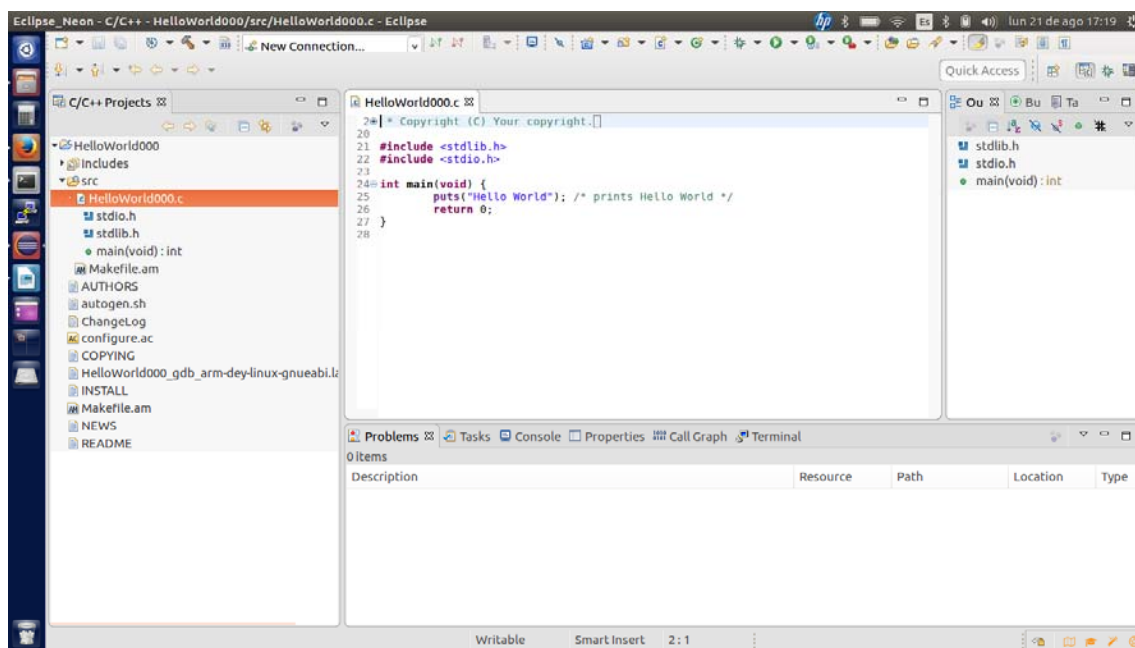


- Presionar **Next** y pasará a una nueva ventana **Basic Settings** en donde como opción podrá cargar datos referentes a Copyright del proyecto.

- Al presionar nuevamente **Next** pasará a la última ventana **Select Configurations** en donde estará tildado por defecto **Buils (GNU)** y **Debug (GNU)**, presionar **Finish**.

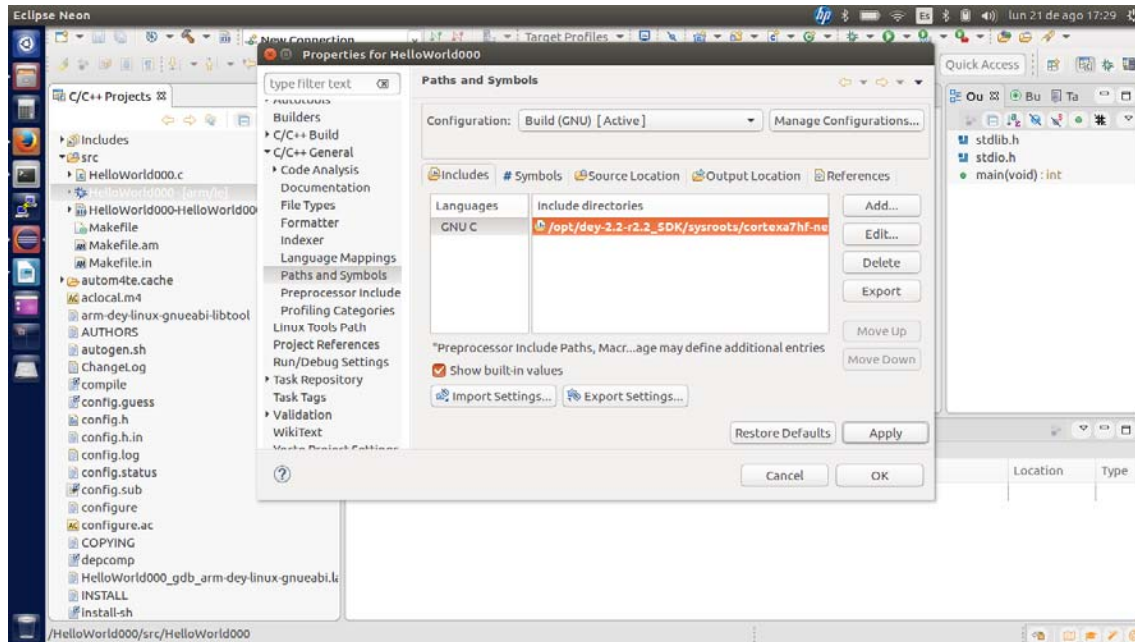


- En el árbol de navegación de la izquierda C/C++ Projects hacer doble click al archivo **HelloWorld000→src→HelloWorld000.c**.



- Abrir **Project→Properties**.

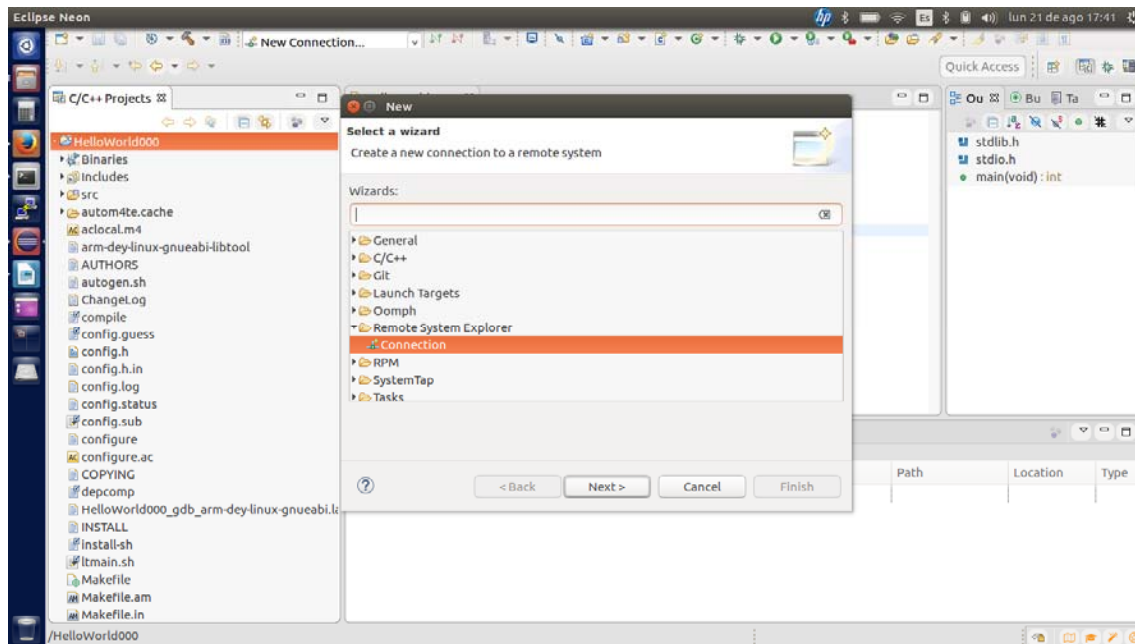
- Dentro de la ventana y en el árbol de la izquierda ir a **C/C++→Path and Symbols**:



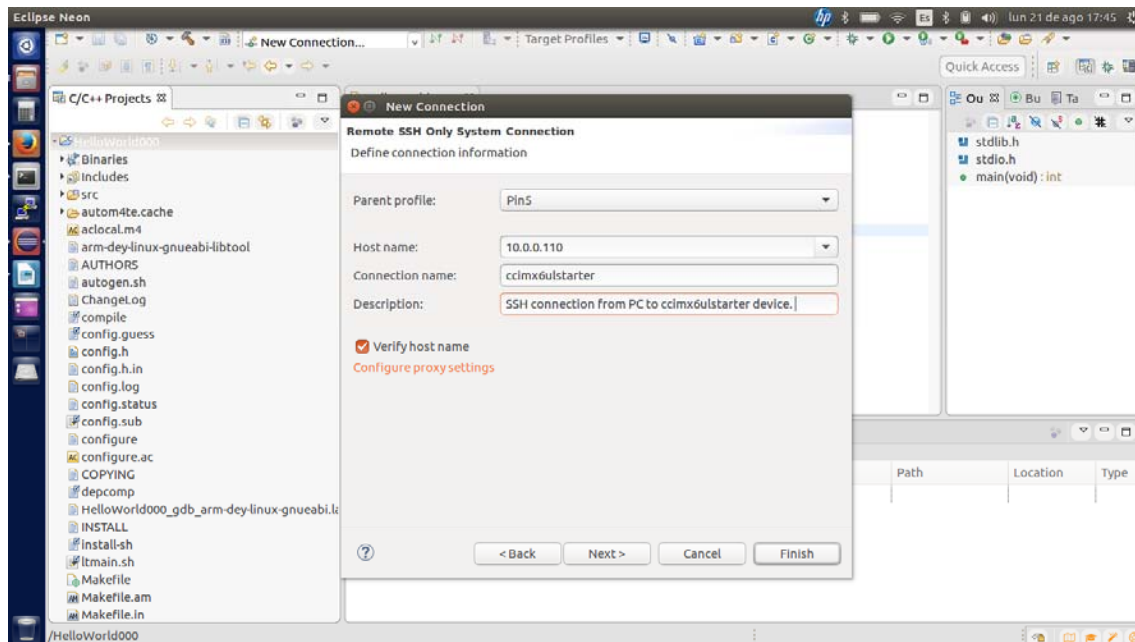
- Presionar el botón **Add...** y agregar como **File System...** los headers de C/C++ para poder luego compilar las aplicaciones: **/opt/dey-2.2-r2.2_SDK/sysroots/cortexa7hf-neon-dey-linux-gnueabi/usr/include**.

6) Creación de una conexión SSH entre Eclipse Neon y el módulo ccimx6ulstarter:

- Configuraremos la conexión a la placa ccimx6ulstarter ingresando en **File→New→Other**:
- Dentro de la ventana seleccionar **Remote System Explorer→Connection**:



- Presione el botón **Next** y elija la opción **SSH Only**:

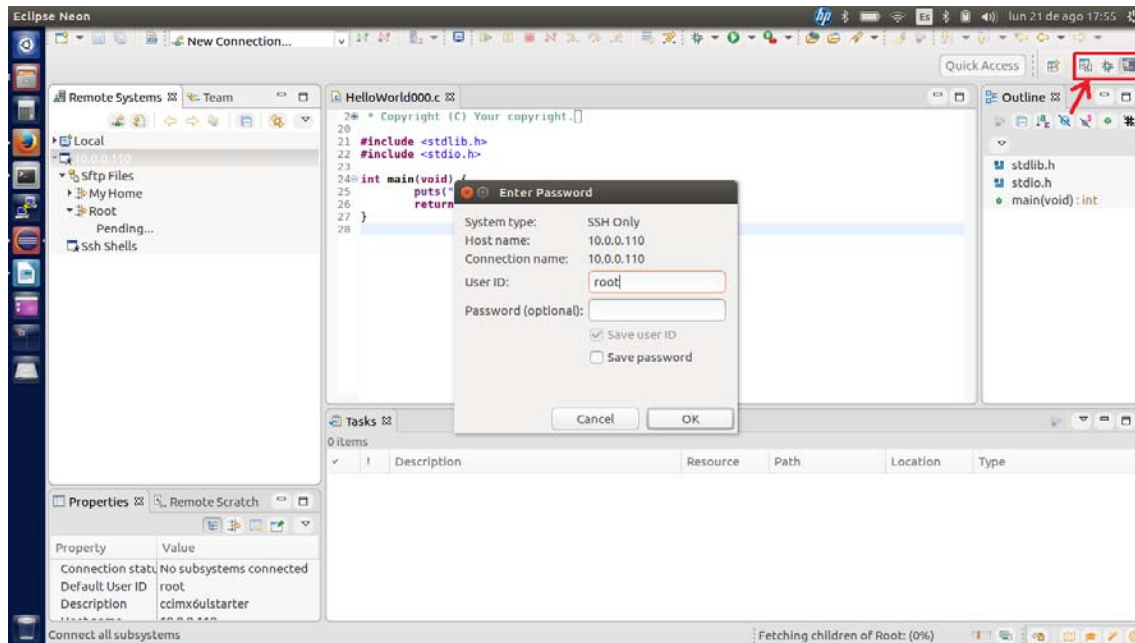


En este paso seleccionar en el textbox “**Host name**” la dirección IP a la que el módulo ccimx6ulstarter se encuentra conectado.

Asimismo crear un nombre en el campo “**Connection name**” y en forma opcional puede rellenar el campo “**Description**” con una descripción adecuada.

- Presione **Finish** para crear la conexión entre la PC y el módulo.

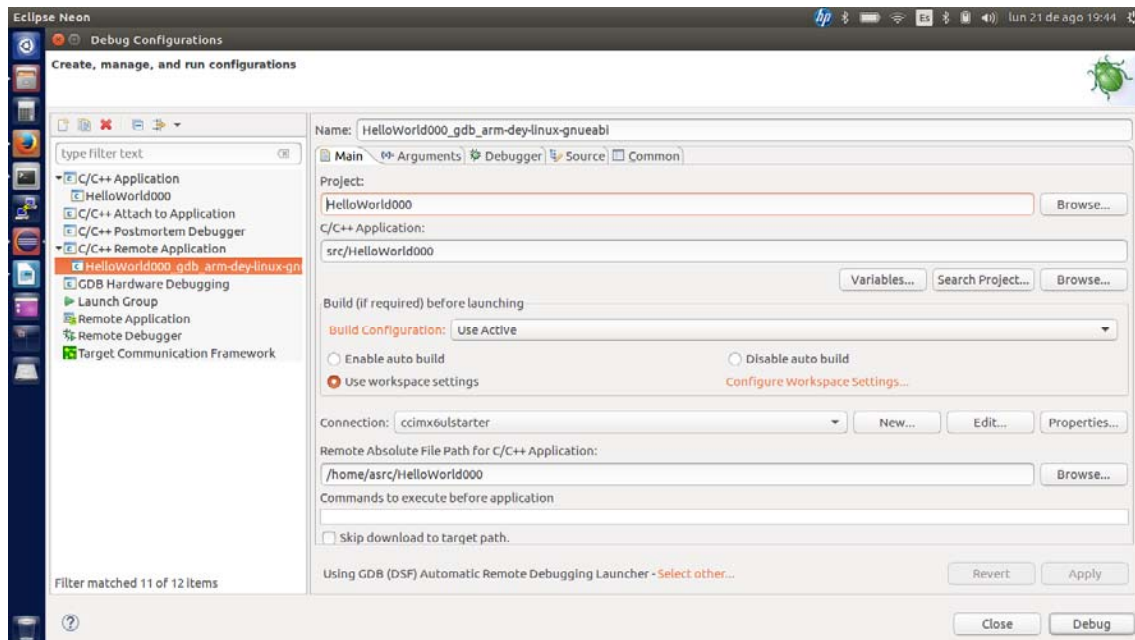
- Una vez finalizada la creación de la conexión y después de compilar por primera vez, en la margen derecha de pantalla principal de Eclipse Neon, aparecerán tres nuevas perspectivas: **C/C++**, **Debug** y **Remote System Explorer**:



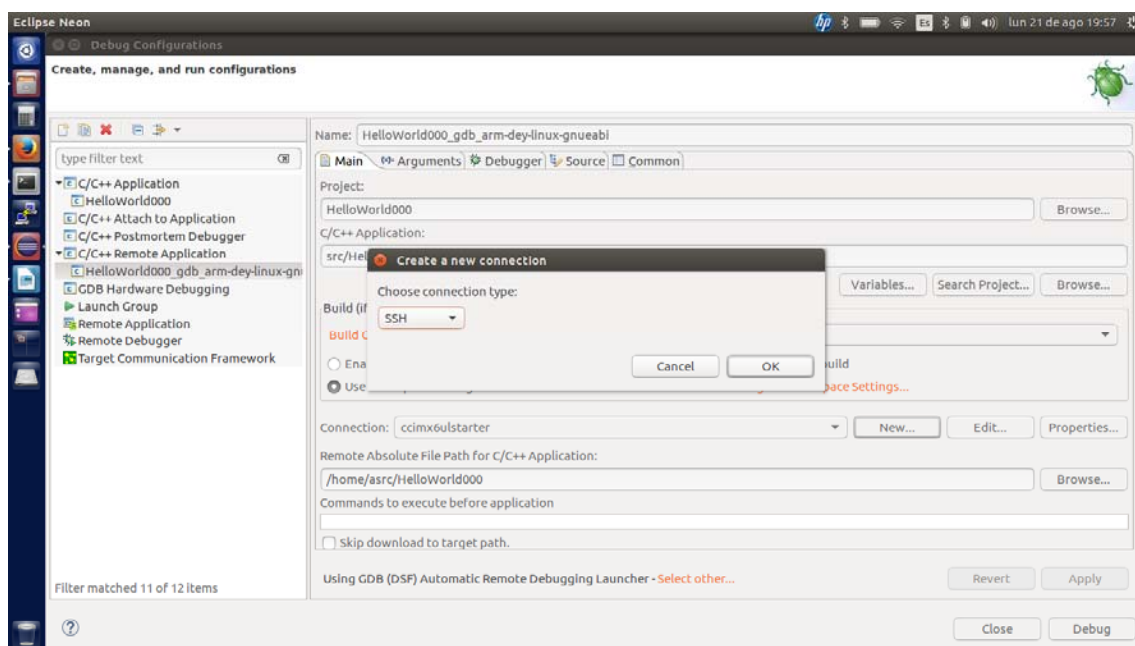
- Luego ingresar a la perspectiva “**Remote System Explorer**” para conectar Eclipse Neon al módulo ccimx6ulstarter, una nueva ventana le pedirá ingresar “**User ID**” y “**Password**”.

7) Configuración del debugger:

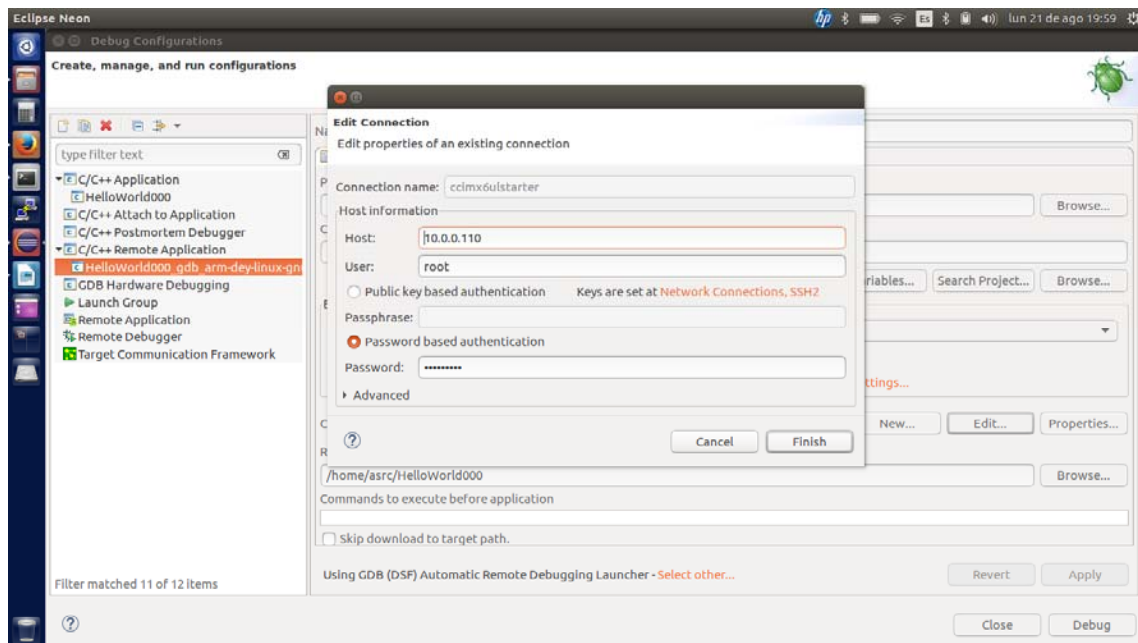
- Previamente los paquetes **gdb** y **gdbserver** deberán estar instalados y configurados correctamente en su módulo ccimx6ulstarter.
- Ingresar de la siguiente manera al menú de configuración del debugger **Run→Debug Configurations**, aparecerá una nueva ventana.



- En la misma deberá realizar la configuración según lo indicado en el print screen previo.
- Luego, en el sector “**Connections**” presionar el botón **New...** para configurar una nueva conexión:



- Seleccionar la opción **SSH** y a continuación completar los datos requeridos en la pantalla de configuración de la conexión:

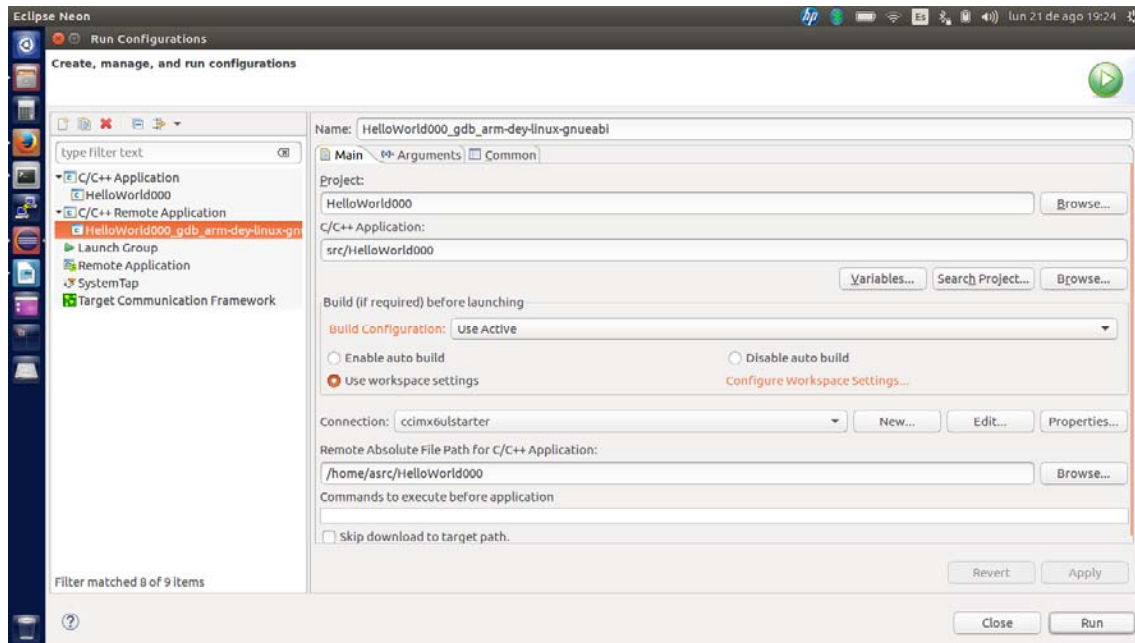


- Los datos a ingresar son:
 - ✓ **Host:** Número IP del dispositivo ccimx6ulstarter.
 - ✓ **User:** Usuario para poder conectarse al módulo, se recomienda configurar un usuario diferente a root y agregar un nuevo password para el mismo para poder incrementar la seguridad.
 - ✓ Seleccionar la opción “**Password based authentication**” y agregar el password de su módulo.
 - ✓ Presionar el botón **Finish**.
- En el textbox de “**Remote Absolute File Path for C/C++ Application**” tipear el directorio dentro del módulo en donde quiera guardar las aplicaciones compiladas.
- Para realizar el debugging remoto de su aplicación presione la tecla situada en la parte inferior derecha de la ventana llama **Debug**.

8) Configurar Run:

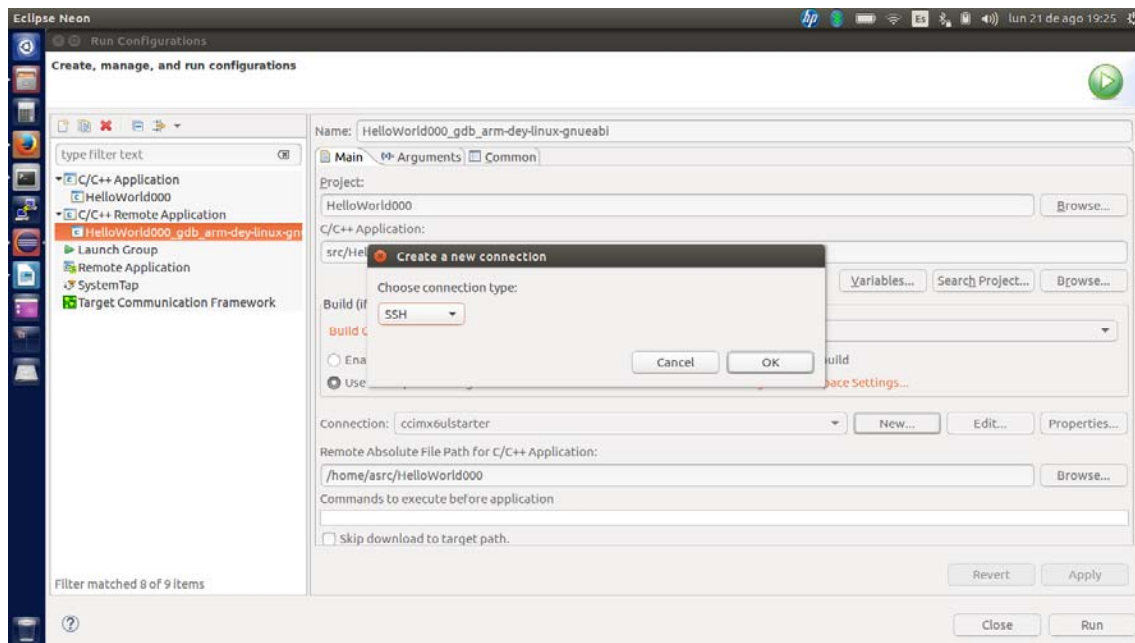
Este punto se utiliza para qué al ejecutar el programa, directamente se vean los resultados en una ventana de consola directamente en la interfaz de desarrollo. Esto acelera mucho las pruebas debido a que puede ejecutar directamente presionando los botones en la perspectiva “**Remote System Explorer**” de Eclipse Neon.

- Ejecutar **Run**→**Run Configurations**:

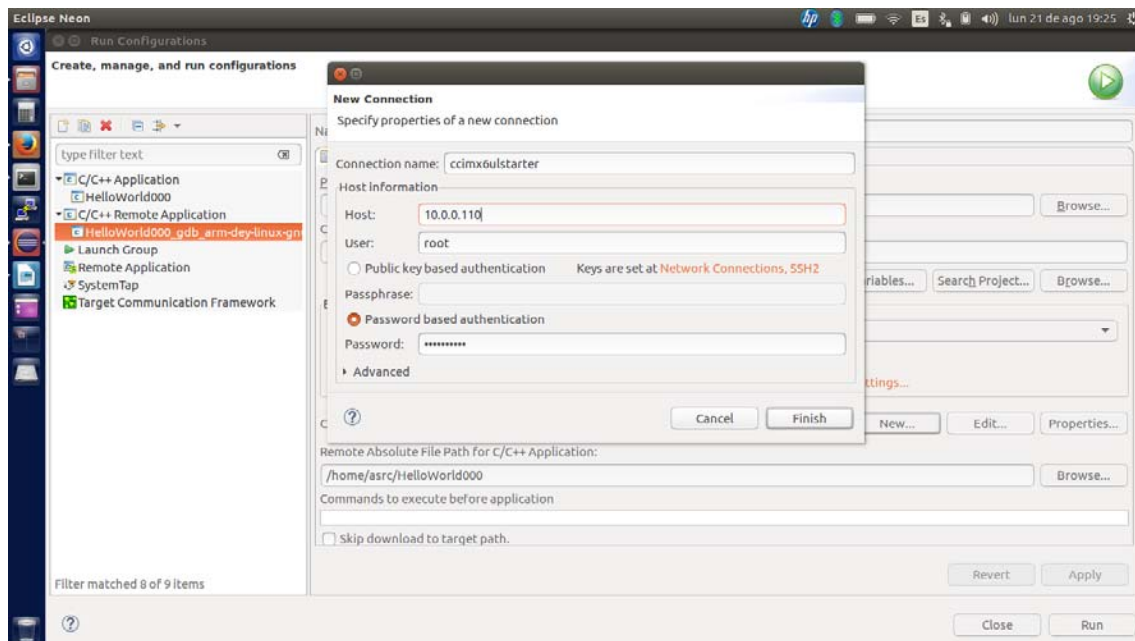


- Si no apareciera en el árbol a la derecha de la ventana la opción **C/C++ Remote Application**→**HelloWorld000_gdb_arm-dey-linux-gnueabi**, deberá crearse cargando los valores que aparecen en la pantalla previa.

- Luego, en el sector “**Connections**” presionar el botón **New...** para configurar una nueva conexión:

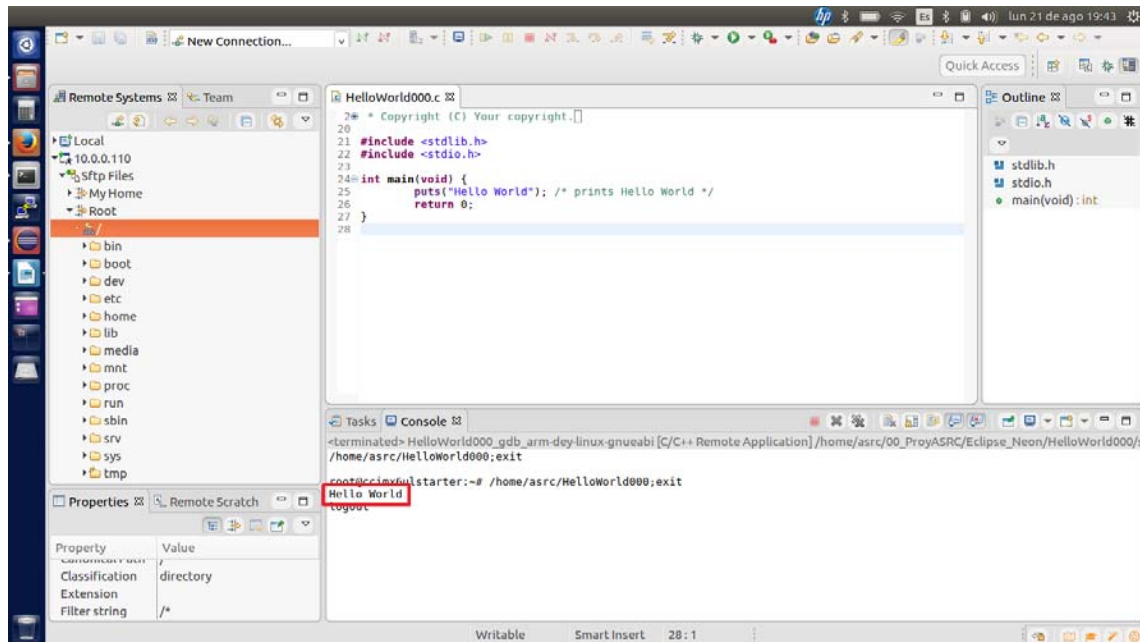


- Seleccionar la opción **SSH** y a continuación completar los datos requeridos en la pantalla de configuración de la conexión:



- Los datos a ingresar son:
 - ✓ **Host:** Número IP del dispositivo ccimx6ulstarter.
 - ✓ **User:** Usuario para poder conectarse al módulo, se recomienda configurar un usuario diferente a root y agregar un nuevo password para el mismo para poder incrementar la seguridad.
 - ✓ Seleccionar la opción “**Password based authentication**” y agregar el password de su módulo.
 - ✓ Presionar el botón **Finish**.

- En el textbox de “**Remote Absolute File Path for C/C++ Application**” tipear el directorio dentro del módulo en donde quiera guardar las aplicaciones compiladas.
- Para realizar la primera prueba de ejecución, y recordando que debe estar sí o sí en la perspectiva “**Remote System Explorer**”, presione la tecla situada en la parte inferior derecha de la ventana llama **Run**.
- Aparecerá una ventana en donde se mostrará una consola en donde podrá ver el inicio de sesión **SSH** en el módulo **ccimx6ulstarter**, la copia al directorio dentro del módulo (en nuestro caso **/home/asrc>HelloWorld000**) del ejecutable de la aplicación que hemos realizado, los resultados de la ejecución y el posterior cierre de la sesión **SSH**.



- Vemos que el ejemplo “**Hello World**” se ha ejecutado satisfactoriamente, esto indica que nuestro IDE Eclipse Neon se encuentra totalmente configurado para programar nuestras aplicaciones.

9) Notas finales:

a) Revisiones del presente documento:

| Revisión | Fecha | Descripción |
|----------|----------------------|--------------------------------|
| 001 | 22 de Agosto de 2017 | Primera versión del documento. |

b) Marcas registradas y copyright:

HomeBrain es marca registrada en la República Argentina. Las demás marcas mencionadas en el presente documento son propiedad de sus respectivos dueños.

c) Descargo de responsabilidad:

La información contenida en este documento está sujeta a cambios sin previo aviso y no representa un compromiso por parte de HomeBrain. HomeBrain proporciona este documento "tal cual" sin garantía de ningún tipo, expresa o implícita, incluyendo, pero no limitado a, las garantías implícitas de aptitud o comercialización para un propósito particular. HomeBrain puede realizar mejoras y/o cambios en este manual o en los productos y/o programas descritos en este manual en cualquier momento.

d) Envíenos sus comentarios:

Para proveer feedback a quienes produjeron el presente documento, envíenos un e-Mail a info@homebrain.com.ar.