

Práctica 1. Instalación de un Entorno de Desarrollo, Compilación Cruzada y Simulación para el Procesador LEON 3

1. Introducción

En este documento se describe el proceso de instalación de un entorno de desarrollo y compilación cruzada para el procesador LEON3. La práctica explica además cómo integrar, en dicho entorno, el control de la ejecución de programas sobre un simulador de este procesador denominado *TSIM2*. LEON3 pertenece a la familia SPARC, y ha sido seleccionado como procesador de referencia por la Agencia Espacial Europea - European Space Agency (ESA) para sus misiones espaciales. Por este motivo forma parte, en numerosos satélites, del sistema empujado denominado *On-Board Data Handling* (OBDH) encargado de gestionar en vuelo los datos de sus distintos subsistemas.

El entorno de desarrollo ha sido instalado sobre el sistema operativo Linux. La distribución concreta sobre la que se ha realizado la instalación ha sido la Ubuntu 14.04 (Trusty Tahr), si bien los pasos a dar pueden considerarse análogos en el caso de que la distribución de Linux elegida fuera otra. Una solución que permite evitar cualquier problemática asociada a la distribución es utilizar una máquina virtual construida a partir de la versión Ubuntu 14.04 como la que se puede descargar del sitio web <http://www.trendsigma.net/vmware/ubuntu1404t.html>. Si se utiliza una instalación de Ubuntu de 64 bits (como ocurre en el puesto de la universidad), la instalación de los paquetes que vienen por defecto deberá completarse de la siguiente forma:

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386
libnsp4-0d:i386 zlib1g:i386 libx11-6:i386
```

En los siguientes apartados se irán enumerado los pasos requeridos para hacer efectiva la instalación, así como algunas pruebas que permiten comprobar que ésta se ha completado correctamente.

2. Instalación del entorno Eclipse

(Si estás en puesto de la universidad, eclipse estará instalado, por lo que sólo tendrás que hacer el punto 5 de este apartado)

Los proyectos C/C++ que se van a crear sobre el procesador LEON3 pueden gestionarse mejor desde un entorno integrado de desarrollo como es Eclipse. Este entorno se puede instalar fácilmente dentro de la distribución ubuntu 14.04 de linux siguiendo los siguientes pasos.

1. Descarga del archivo tar.gz que contiene los archivos del entorno Eclipse configurado para trabajar con proyectos C/C++. Este archivo, para la versión *kepler* de Eclipse, se puede descargar directamente del enlace:

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/kepler/SR2/eclipse-cpp-kepler-SR2-linux-gtk.tar.gz&mirror_id=580

2. Mover el archivo al directorio /opt

```
sudo mv eclipse-cpp-kepler-SR2-linux-gtk.tar.gz /opt
```

3. Descomprimir el archivo en ese mismo directorio

```
cd /opt/  
sudo tar -xzf eclipse-cpp-kepler-SR2-linux-gtk.tar.gz
```

4. Instalar el *run-time Java* que permite ejecutar la aplicación Eclipse. Esta instalación se puede realizar bien utilizando el comando *apt-get*:

```
sudo apt-get update  
sudo apt-get install openjdk-6-jre
```

5. Añadir (utilizando, por ejemplo, gedit o vim) **al final del archivo** /home/user/.profile (o /home/atcsol/.profile, si **estás en el puesto de la universidad**) la siguiente línea de modificación del PATH que incorpora el directorio donde se almacena el ejecutable que lanza el entorno eclipse

```
PATH="/opt/eclipse:$PATH"
```

Si **estás en el puesto de la universidad** la línea es:

```
PATH="/opt/eclipse/eclipsekepler4.3.2:$PATH"
```

3. Instalación del compilador cruzado BCC

El compilador cruzado BCC (Bare-C Cross Compiler) es un compilador cruzado para los procesadores LEON2 y LEON3. La distribución del compilador incluye además la nueva biblioteca de C *Newlib standalone* y rutinas de bajo nivel para entrada y salida sobre LEON2 y LEON3. Finalmente la distribución también proporciona la utilidad Mkprom que permite crear, a partir de un sistema software ejecutable en RAM, un nuevo software ejecutable desde una memoria no volátil (PROM o EEPROM) y cuyo objetivo es únicamente desplegar en RAM el sistema original y darle paso.

La instalación del compilador BCC comprende los siguientes pasos.

1. Descargarse la distribución de BCC sparc-elf-4.4.2-1.0.45.tar.bz2

<http://www.gaisler.com/anonftp/bcc/bin/linux/sparc-elf-4.4.2-1.0.45.tar.bz2>

2. Descomprimirla en el directorio /opt

```
sudo tar -C /opt -xjf sparc-elf-4.4.2-1.0.45.tar.bz2
```

3. Crear un enlace blando con el nombre `sparc-elf`

```
cd /opt
sudo ln -s sparc-elf-4.4.2 sparc-elf
```

4. Añadir **al final del archivo** `/home/user/.profile` (o `/home/atcsol/.profile`, **si estás en el puesto de la universidad**) la siguiente línea de modificación del PATH que incorpora el directorio donde se almacenan los archivos binarios del compilador BCC

```
PATH="/opt/sparc-elf/bin:$PATH"
```

4. Instalación del simulador TSIM2

El simulador TSIM2 permite simular el procesador LEON3. Su instalación comprende los siguientes pasos.

1. Descargarse el simulador de LEON3 `tsim-eval-2.0.41.tar.gz`

<http://www.gaisler.com/anonftp/tsim/tsim-eval-2.0.41.tar.gz>

2. Descomprimirla en el directorio `/opt`

```
sudo mv tsim-eval-2.0.41.tar.gz /opt
cd /opt
sudo tar -xzf tsim-eval-2.0.41.tar.gz
```

5. Crear un enlace blando con el nombre `tsim`

```
sudo ln -s tsim-eval/tsim/linux tsim
```

6. Añadir **al final del archivo** `/home/user/.profile` (o `/home/atcsol/.profile`, **si estás en el puesto de la universidad**) la siguiente línea de modificación del PATH que incorpora el directorio donde se almacenan el archivo binario del simulador `tsim`.

```
PATH="/opt/tsim:$PATH"
```

7. Forzar la ejecución del script `./profile` con la orden:

```
bash -l
```

8. Mostrar el PATH actual y comprobar que aparecen los directorio `/opt/tsim` , `/opt/sparc-elf/bin` y `/opt/eclipse`.

```
echo $PATH
```

9. Lanzar el simulador `tsim-leon3`

```
tsim-leon3
```

10. Lanzar el test `paranoia` dentro del simulador `tsim-leon3`, y comprobar que aparece el mensaje “Program exited normally”

```
load /opt/tsim-eval/tests/paranoia
run
```

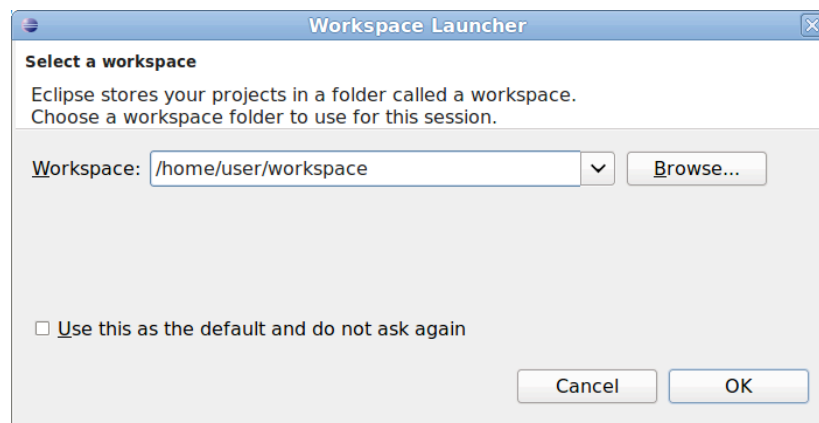
11. Emplear la orden `quit` para salir del simulador `tsim-leon3`.

5. Instalación del plugin de Eclipse para la integración del compilador cruzado BCC y el simulador TSIM2

1. Ejecutar eclipse desde el terminal mediante la orden:

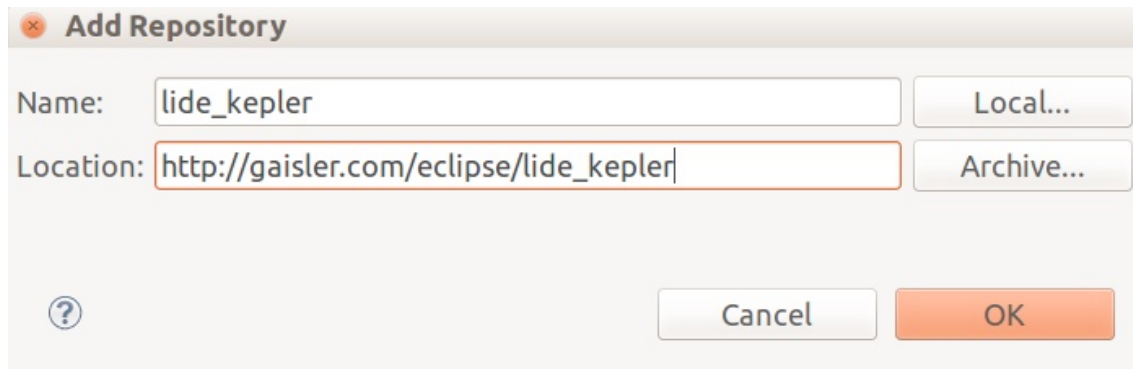
```
eclipse
```

2. Aceptar el nombre del workspace que propone la siguiente ventana inicial de eclipse:

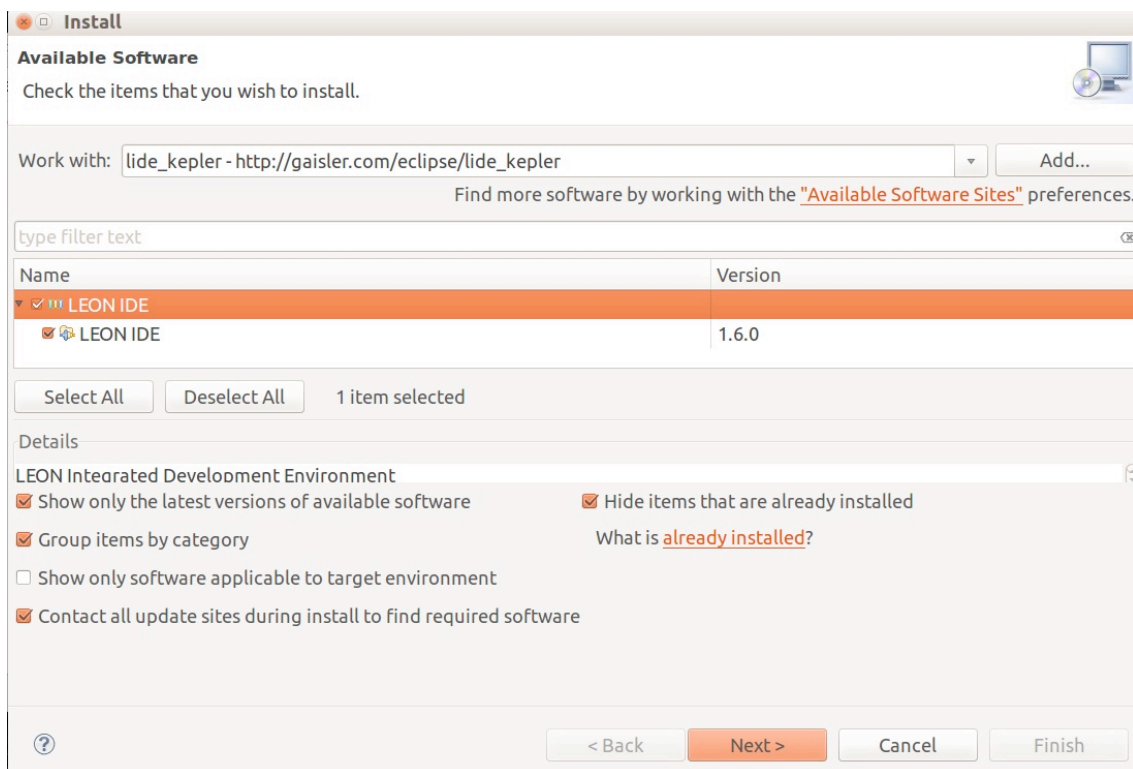


Para integrar el compilador y el simulador con el entorno de desarrollo Eclipse de forma sencilla utilizaremos el plugin de Gaisler denominado CDT Plugin 1.6.0 La instalación de este plugin comprende los siguientes pasos:

3. Seleccionar el menú de Eclipse **Help->Install New Software**
4. Añadir un nuevo repositorio con el boton de la parte derecha **Add...**, e incluir **lide_kepler**, utilizando la URL http://gaisler.com/eclipse/lide_kepler

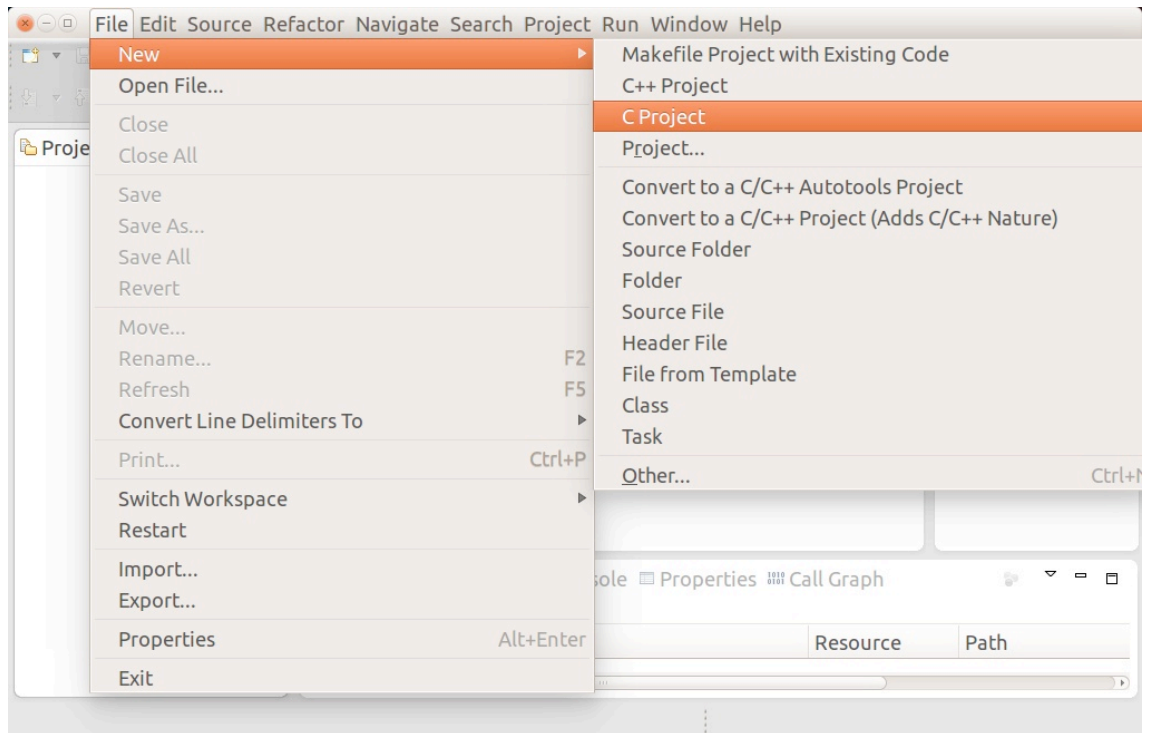


5. Seleccionar LEON IDE, como aparece en la figura siguiente, y aceptar el resto de menus, incluido el que finalmente fuerza el reinicio de eclipse.

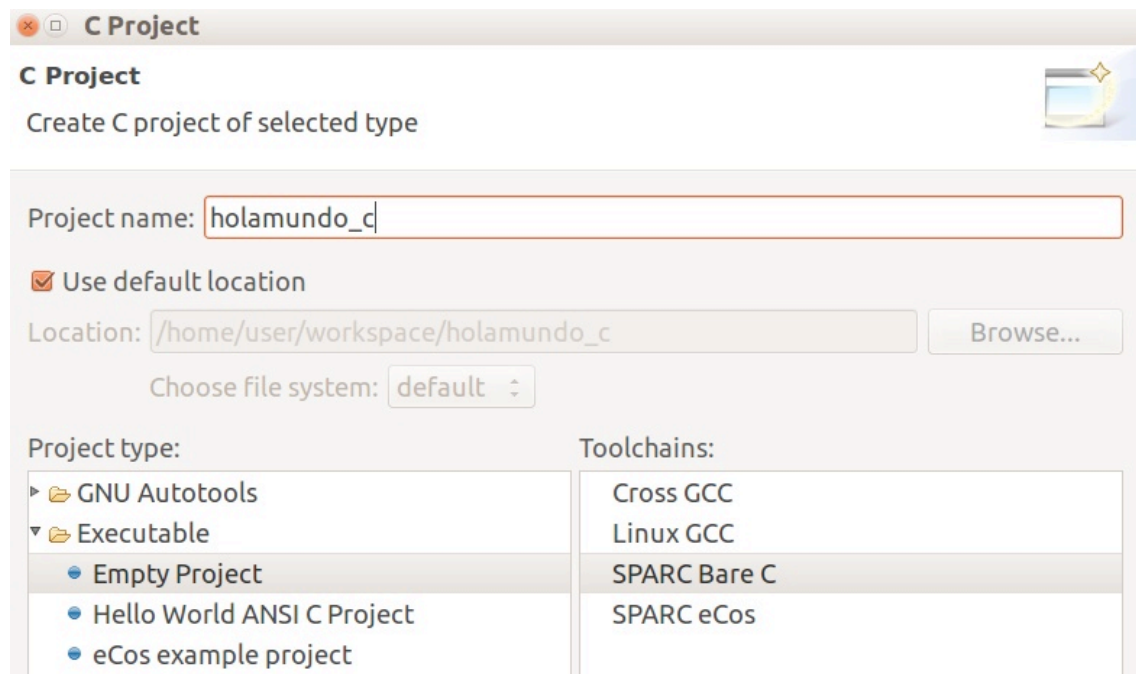


6. Creación un proyecto BCC

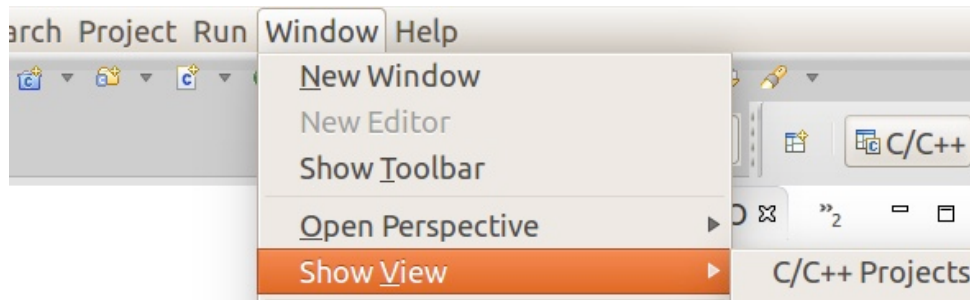
Para crear un proyecto C para el compilador cruzado BCC en el que el Makefile se genere automáticamente hay que seleccionar el menu **File->New-> C Project** tal como se muestra en la figura.



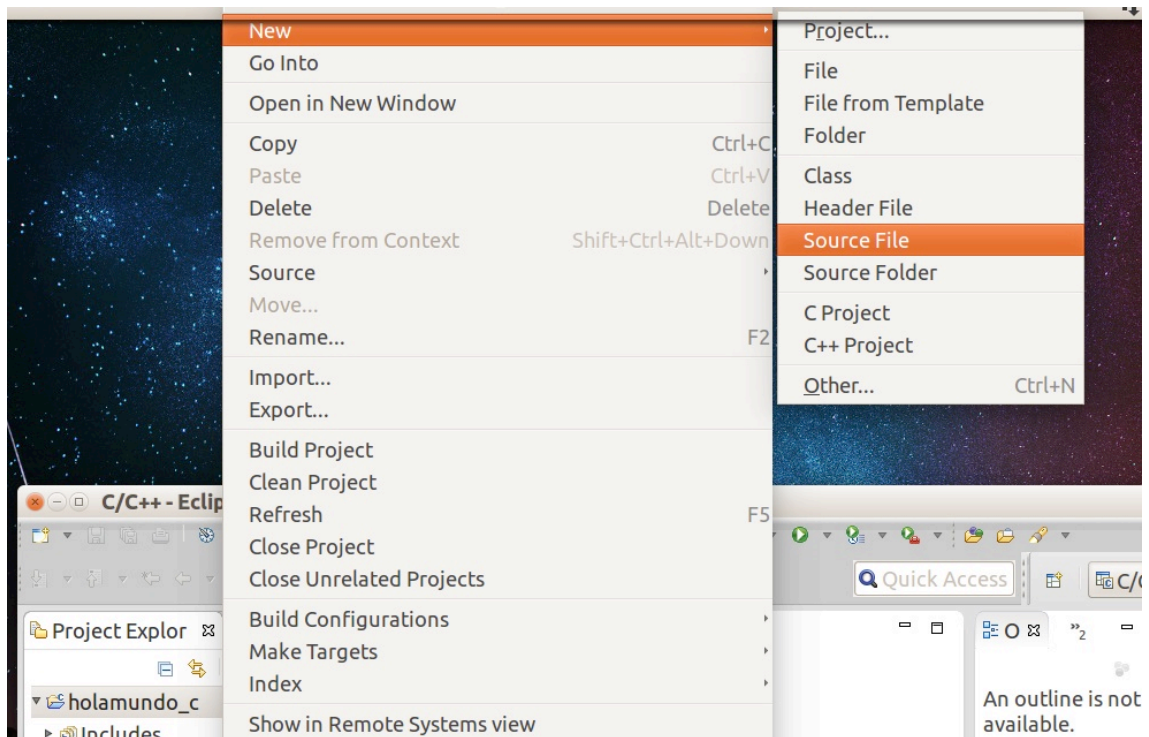
A continuación hay que asignarle un nombre al proyecto (`holamundo_c` en este caso) y definir el tipo de proyecto como **Executable Empty Project** y como *Toolchain Sparc Bare C*, para que se utilice el compilador cruzado de la arquitectura Sparc, a la que pertenecen los procesadores LEON.



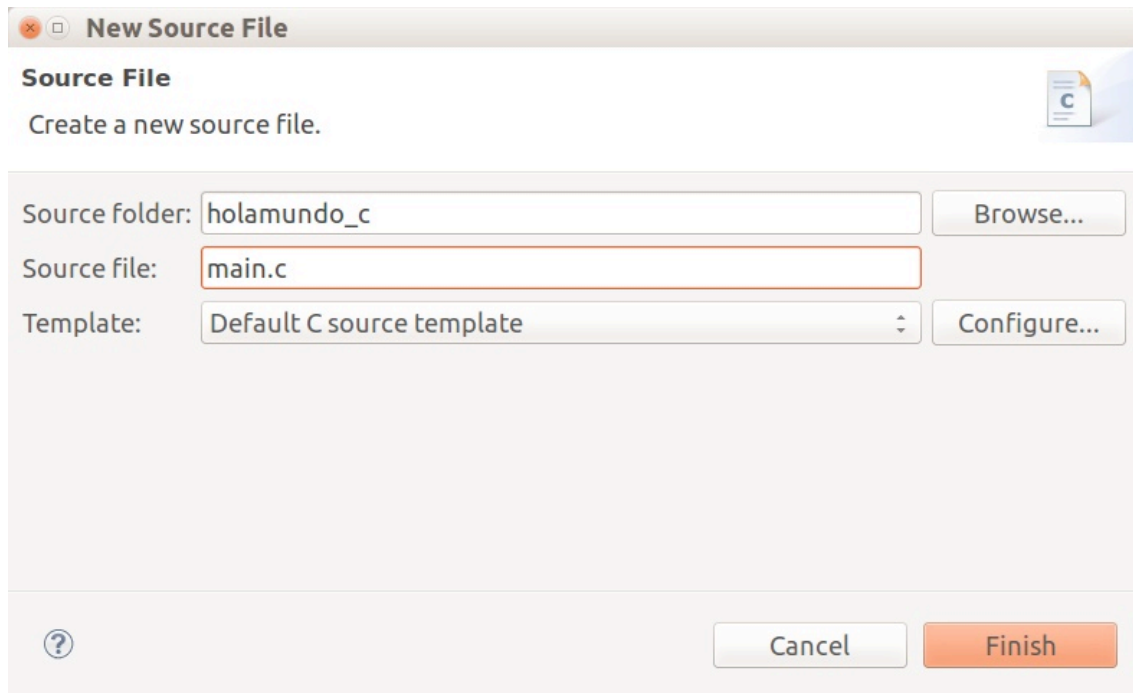
Una vez creado el proyecto, seleccionar la vista C/C++ utilizando el menu **Window->ShowView> C/C++ Projects**



y Añadimos un archivo fuente utilizando el botón derecho sobre el proyecto holamundo_c



, y le damos el nombre **main.c**



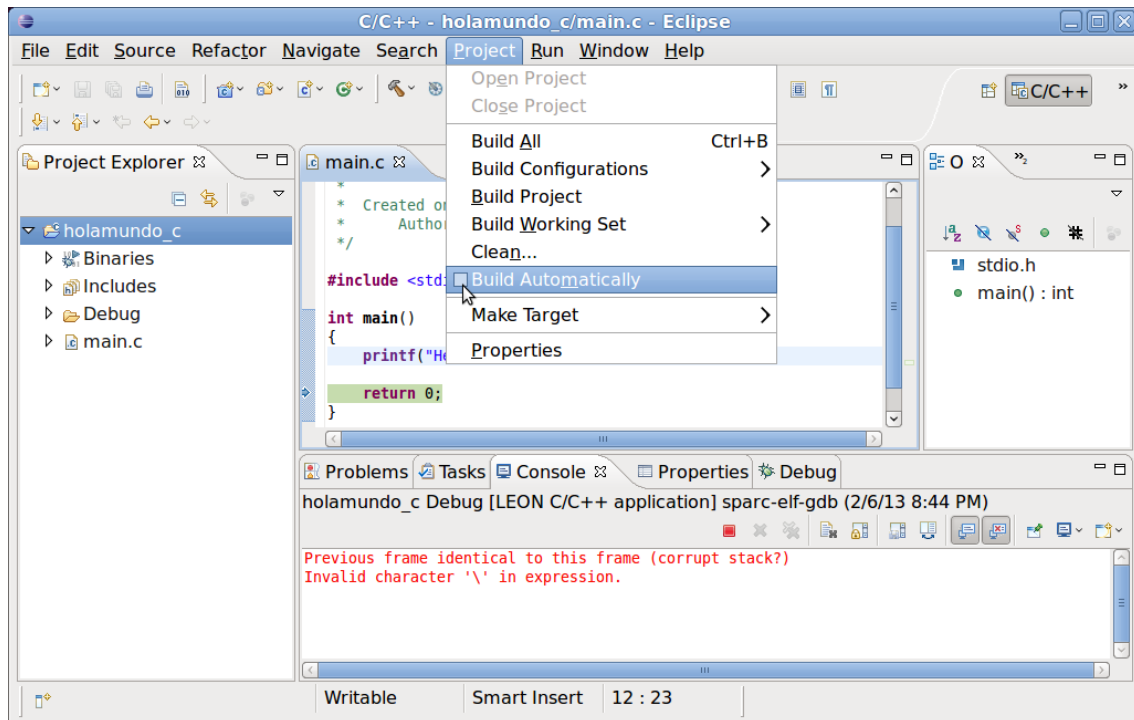
Haciendo doble clic en este archivo añadimos un código básico de hola mundo

```
#include <stdio.h>

int main()
{
    printf("Hello\n");

    return 0;
}
```

El ejecutable de este proyecto se crea automáticamente según la configuración inicial de Eclipse. Esta configuración se puede modificar desde el menú **Project** desmarcando la opción **Build Automatically**.

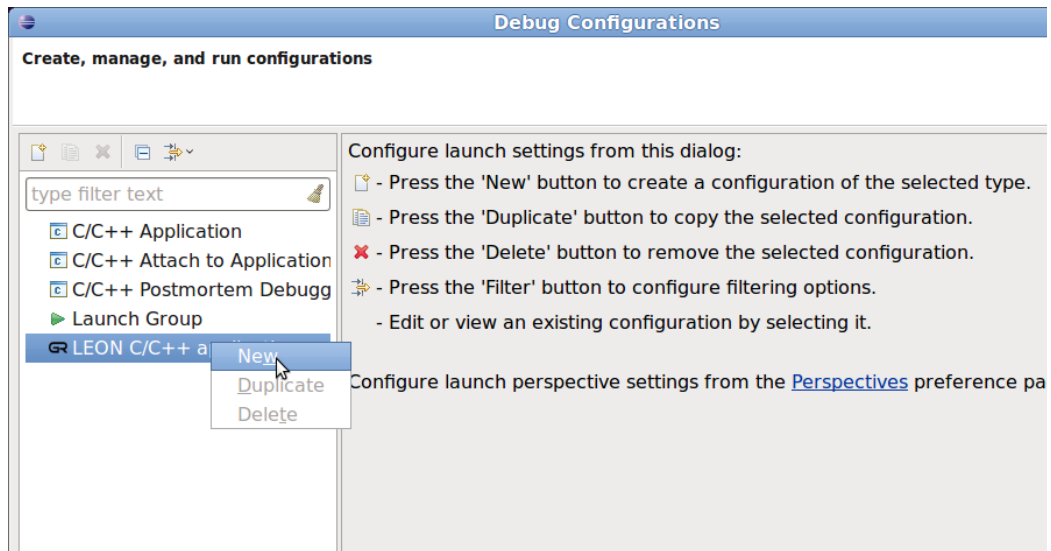


En el caso de que la configuración automática esté deshabilitada, se puede construir el ejecutable mediante el menú **Project->Build Project**

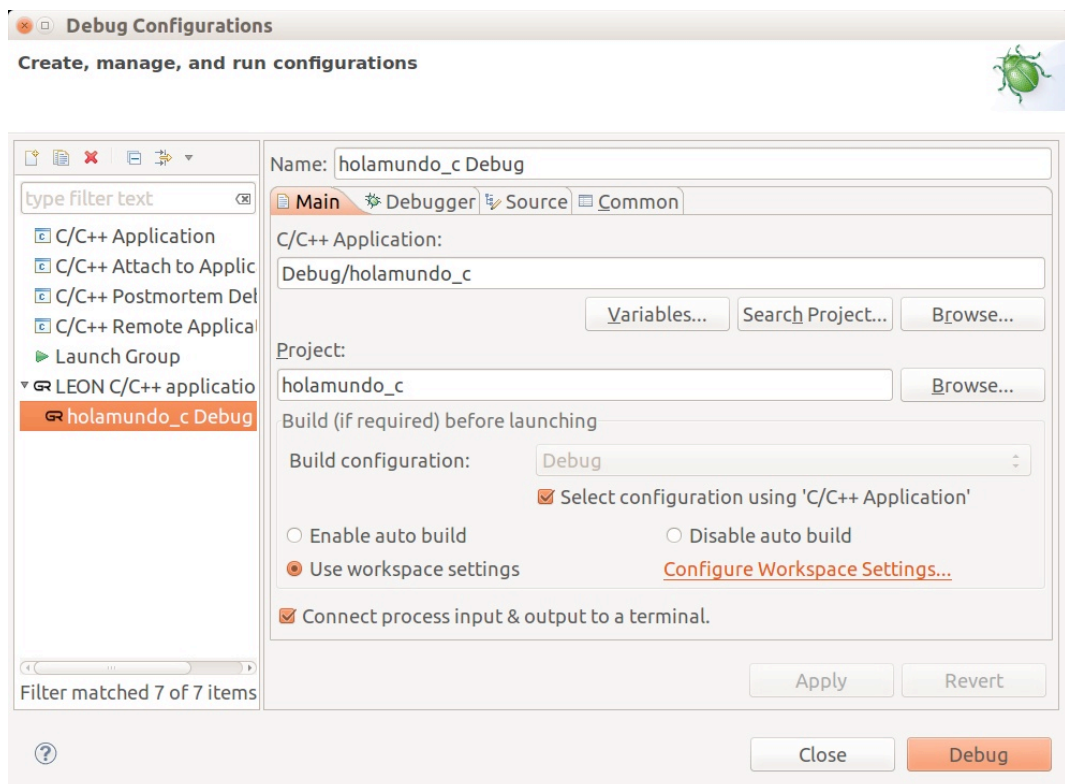
7. Ejecutar en Depuración el proyecto sobre el simulador TSIM2

Para ejecutar el programa, es necesario crear un lanzador que se configure para trabajar con el simulador TSIM2. Para ello habrá que dar los siguientes pasos:

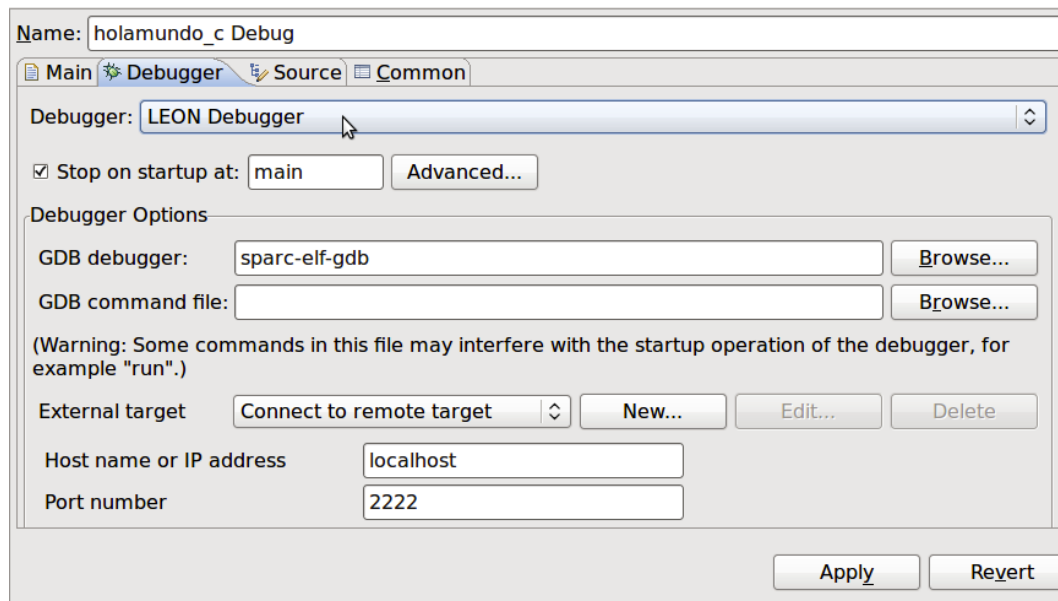
1. Acceder al menú de configuración de depuración mediante el menú **Run->Debug Configurations...**
2. Crear un nuevo lanzador para una aplicación LEON C/C++



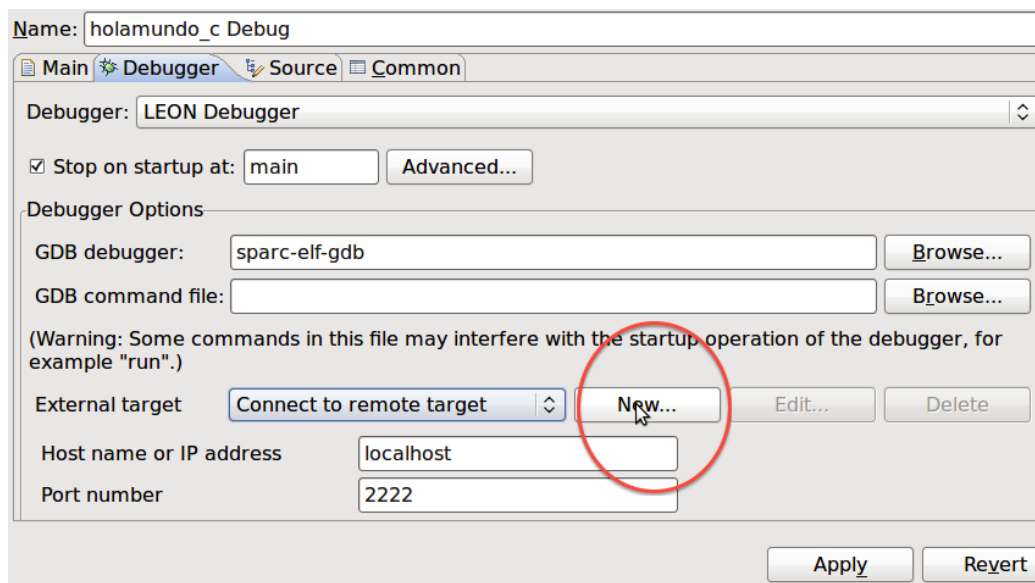
3. Comprobar que la configuración **Main** se corresponde con la siguiente:



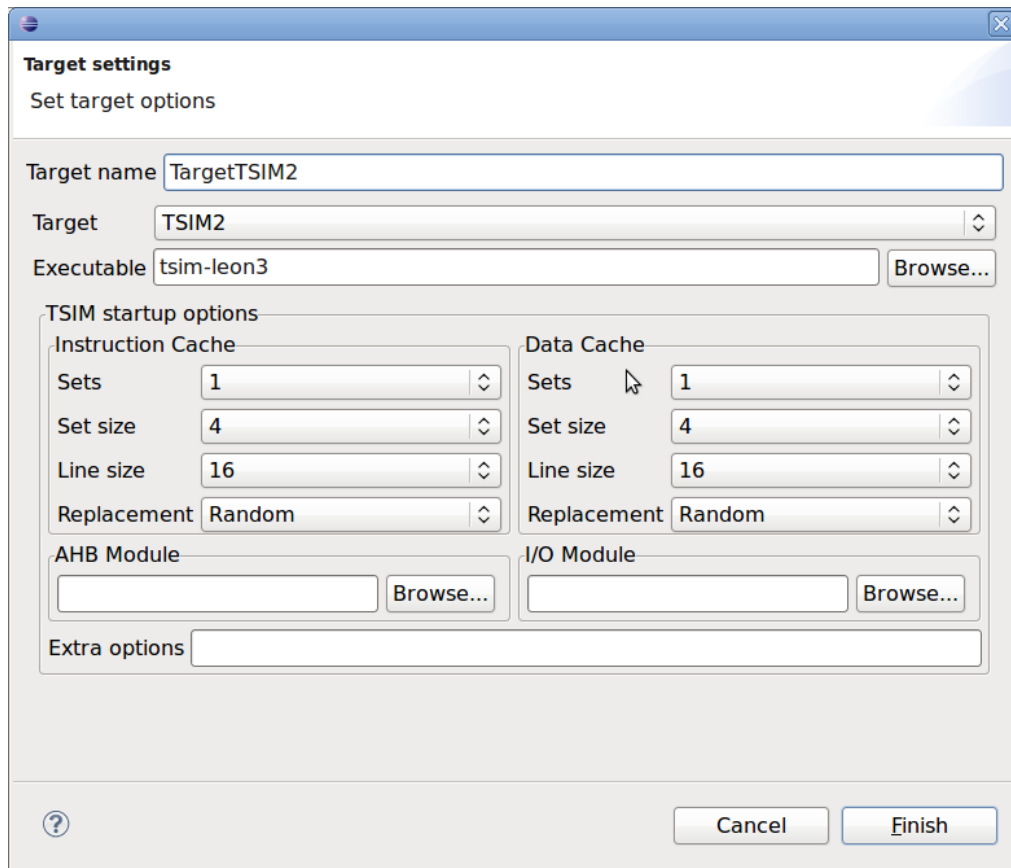
4. Seleccionar **LEON Debugger** como **Debugger** y **sparc-elf-gdb** en la opción de depuración **GDB debugger**:



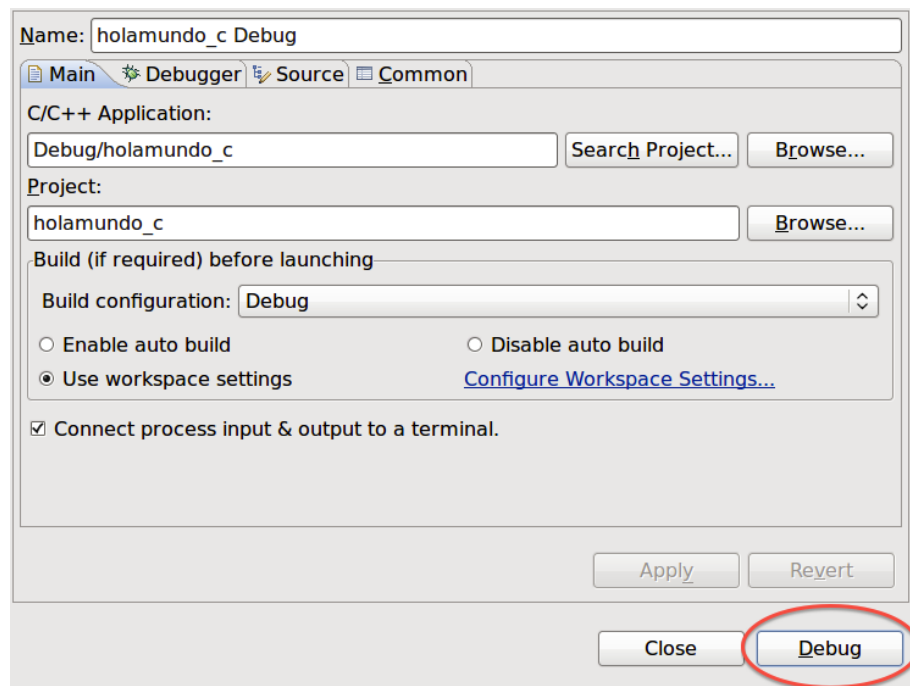
5. Crear un nuevo **External target** (**New...**) que asociaremos al simulador TSIM2.



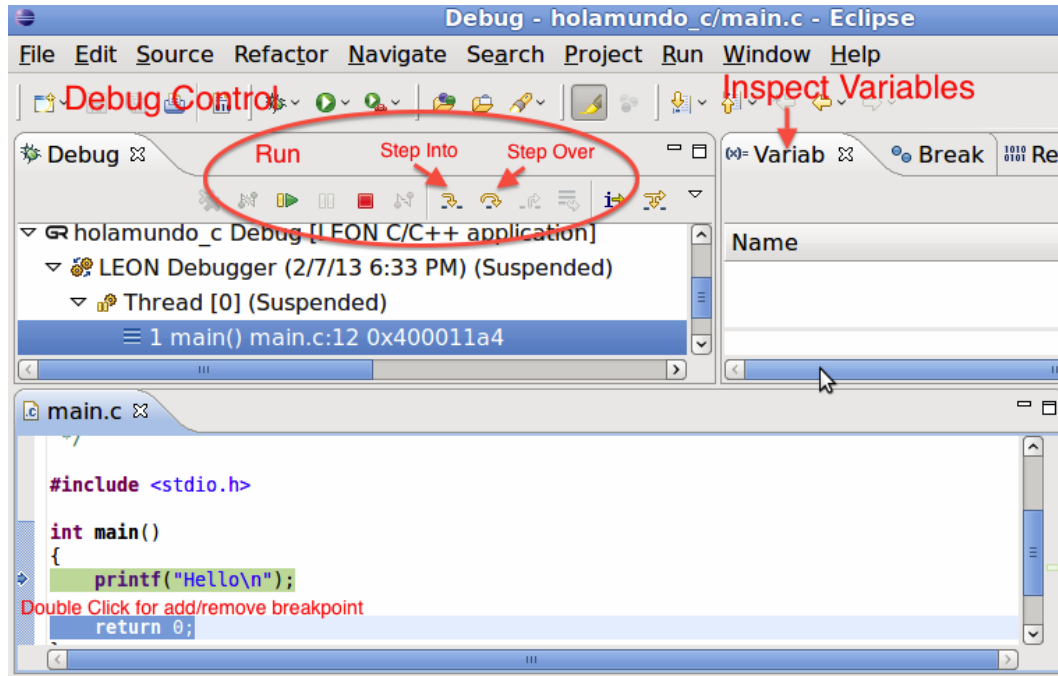
6. Fijamos el **Target name** como **TargetTSIM2**, asociado al **Target TSIM2** y cuyo **Executable** es **tsim-leon3**.



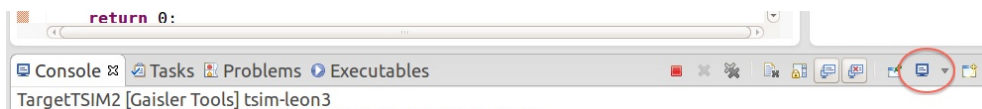
7. Por último lanzar la depuración utilizando el botón **Debug**



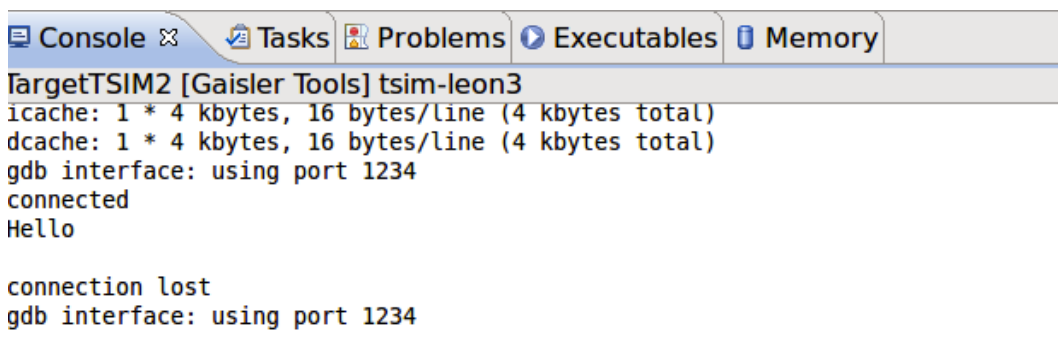
8. Desde la vista **Debug** se puede controlar el lanzamiento de la ejecución continua (**Run**), la ejecución paso a paso (**Step Into** y **Step Over**), la edición de breakpoints (**Double Click for add/remove breakpoint**) y la inspección de variables (**Inspect Variables**).



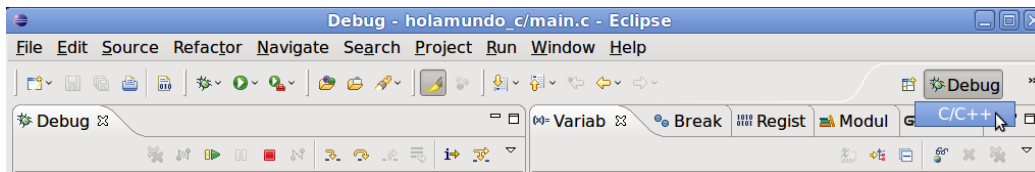
9. Ejecutar el programa seleccionando el control **Run** y seleccionar la consola TargetTSM2 utilizando el elemento marcado con una circunferencia en el panel.



10. Comprobar en la consola que aparece el mensaje “Hello”. Comprobar además que el control de la depuración se hace a través de **gdb** usando el puerto 1234 sobre el localhost.



11. Volver a la Perspectiva C/C++



12. Editar el archivo fuente **main.c** y crear un bucle que muestre por pantalla 10 veces el mensaje *Hello*.

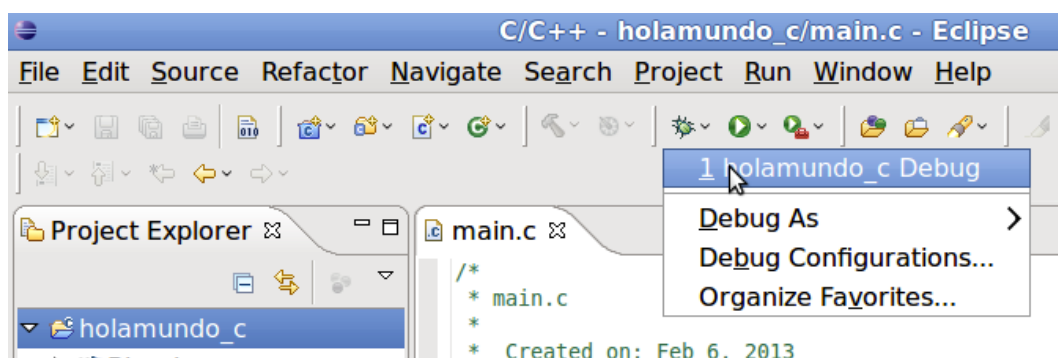
```
#include <stdio.h>

int main()
{
    int i;
    for(i=0; i< 10; i++)
        printf("Hello\n");

    return 0;
}
```

13. Salvar el archivo e invocar **Build->Project** para construir el ejecutable.

14. Lanzar la depuración del nuevo ejecutable utilizando el siguiente menú desplegable.



15. Añadir un **breakpoint** sobre la línea printf y lanzar la ejecución (**Run**) comprobando como en el panel de inspección de variables varía el valor de i en cada iteración del bucle.

