Diseño de Metrónomo con la placa STM32F4-Discovery.

• Herramientas hardware y software para probar el ejercicio.

En éste ejercicio práctico se ha desarrollado software en C para implementarlo en el micro STM32F407VGt6 de la placa de desarrollo STM32F4-Discovery.

Para programar el micro STM32F407VGt6 de la placa de desarrollo STM32F4-Discovery, yo suelo usar o bien el IDE Keil µVIsion5 o el **IDE SW4STM32** (System Workbench for STM32), estando ésta última basada en Eclipse.

En éste ejercicio en particular se ha usado el IDE SW4STM32.

Para instalar dicho IDE dejo el siguiente link:

http://www.ac6-tools.com/downloads/SW4STM32/

Ahí se descarga la versión acorde al sistema operativo que use y la versión que desee.

Yo descargué la versión para Windows de 56 bits:

install_sw4stm32_win_64bits-v1.8

Para ello dentro de la página http://www.ac6-tools.com/downloads/SW4STM32/ puede hacer clic en:

install_sw4stm32_win_64bits-v1.8.exe

o en:

install_sw4stm32_win_64bits-v1.8.zip

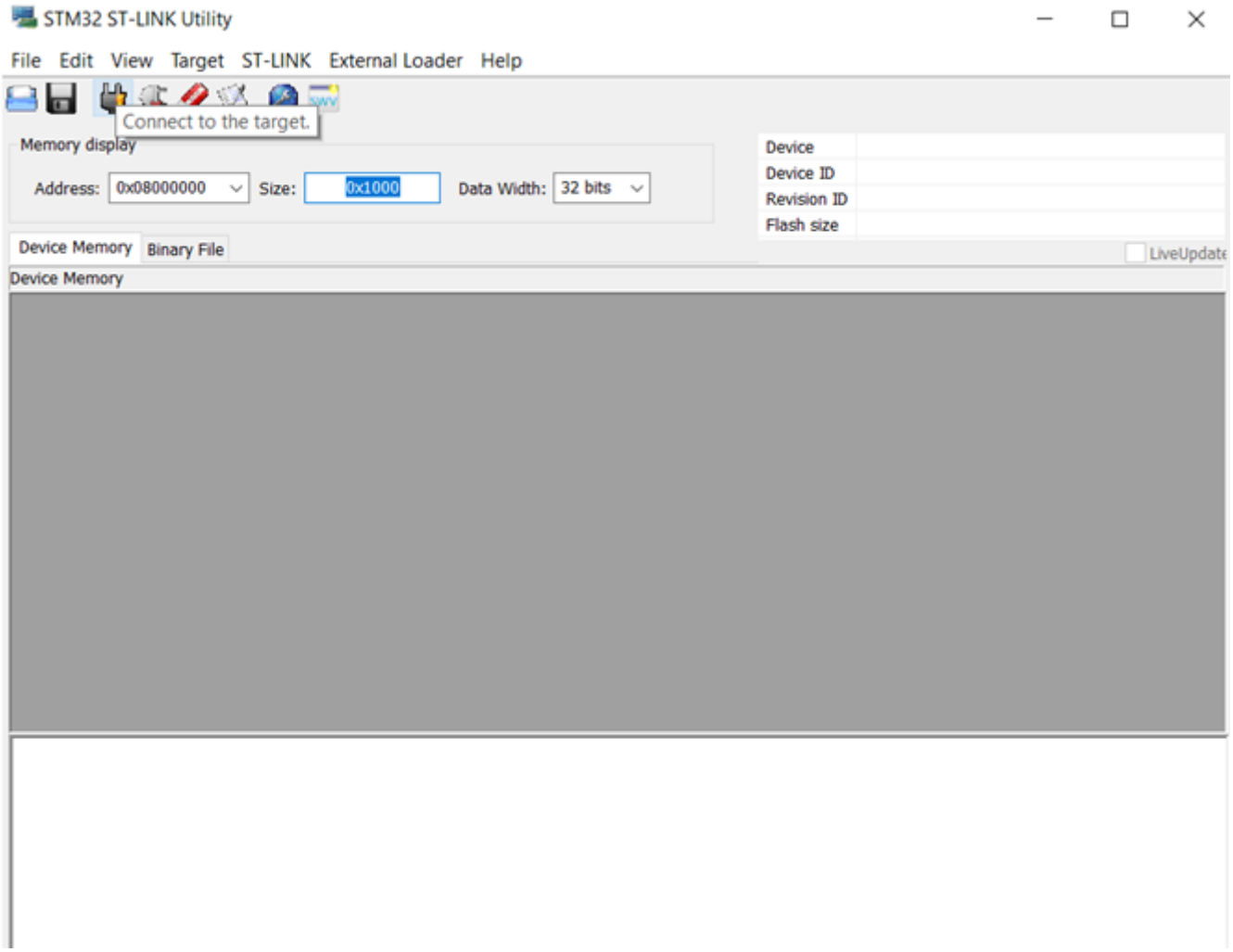
Puede descargar el archivo en la carpeta que desee y navegar a dicha carpeta para proceder a la instalación del IDE.

Algo más que se necesita es el **STM32 ST-LINK utility**, este software nos permite conectarnos con la placa vía USB (entre otras corsas que para probar el ejercicio no son necesarias), para descargarlo dejo el siguiente link:

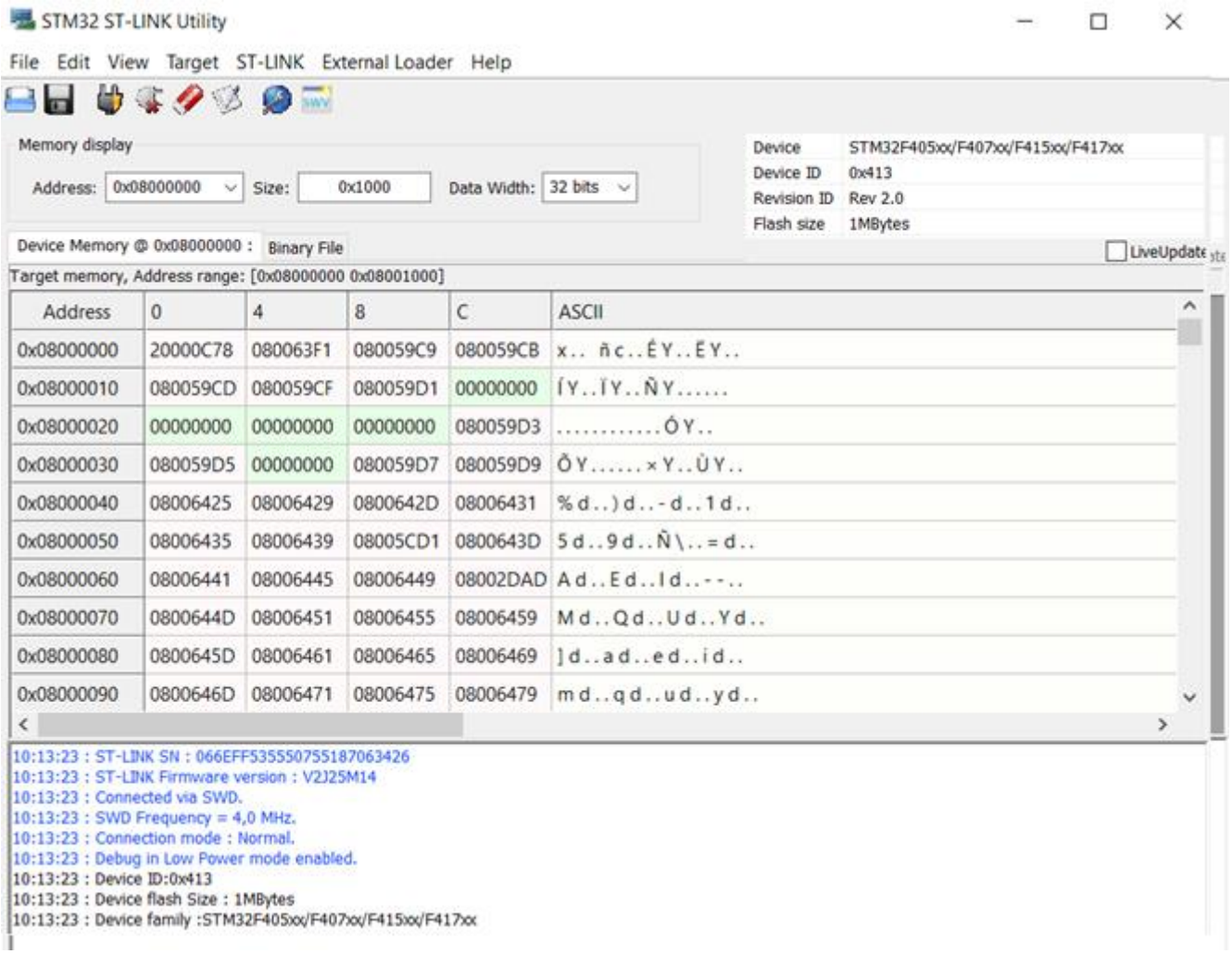
https://www.st.com/en/development-tools/stsw-link004.html

Se instala el STM32 ST-LINK utility (si es que no lo tiene ya instalado).

A continuación, probaremos la conexión con la placa mediante el conector USB. Conectamos la placa al ordenador y abrimos el programa STM32 ST-Link. La pantalla siguiente debe mostrarse a continuación:



Si hacemos clic en el botón señalado (Connect to the target) vamos a realizar la primera conexión entre el microcontrolador y el ordenador. La pantalla deberá cambiar a la siguiente:



En este punto tenemos el sistema ST-Link/V2-A conectado al ordenador y podemos empezar a realizar un desarrollo sobre el sistema.

La herramienta STM32 ST-LINK Utility nos puede servir entre otras cosas para:

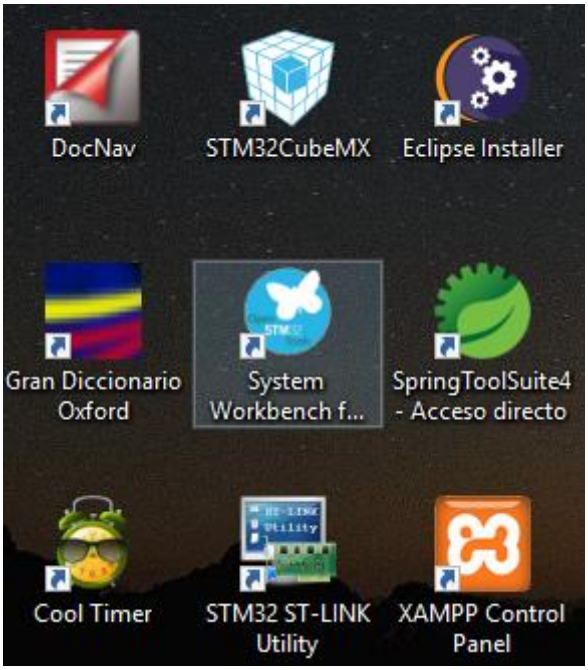
- Programar el microcontrolador (Menú Target/Program).
- Borrar el microcontrolador.
- Visualizar el estado e identificadores del microcontrolador.

Para desarrollar éste ejercicio se usó el STM32CubeMX del que hay bastantes tutoriales e información en Internet. Por si el lector está interesado en usarlo dejo su enlace de descarga:

https://www.st.com/en/development-tools/stm32cubemx.html

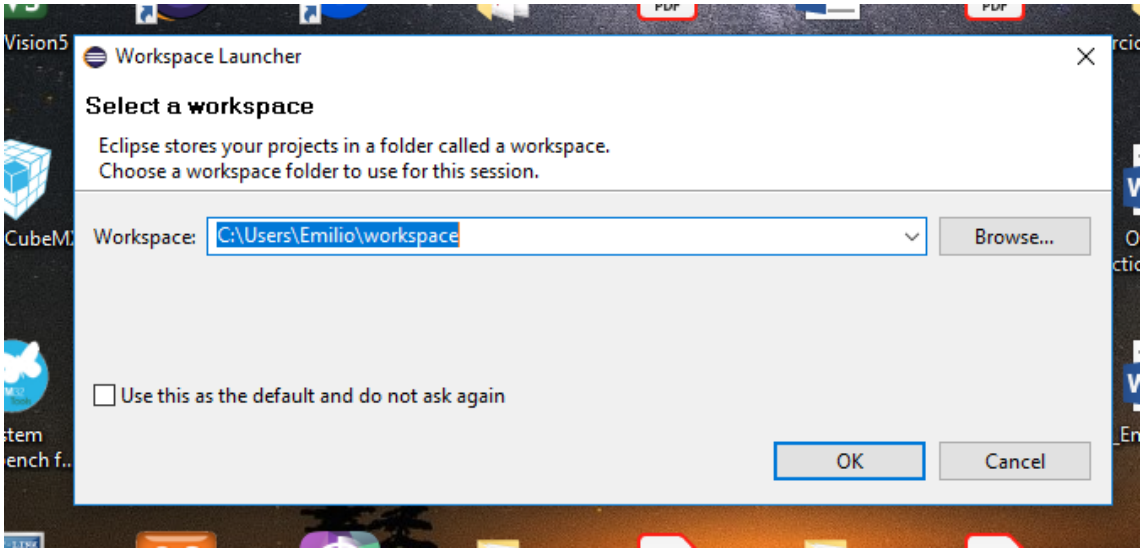
• Abrir el programa desde el IDE SW4STM32 (System Workbench for STM32):

Ya instalado el **IDE SW4STM32**, lo abrimos haciendo clic en su icono en el escritorio o de la forma en que el lector desee:

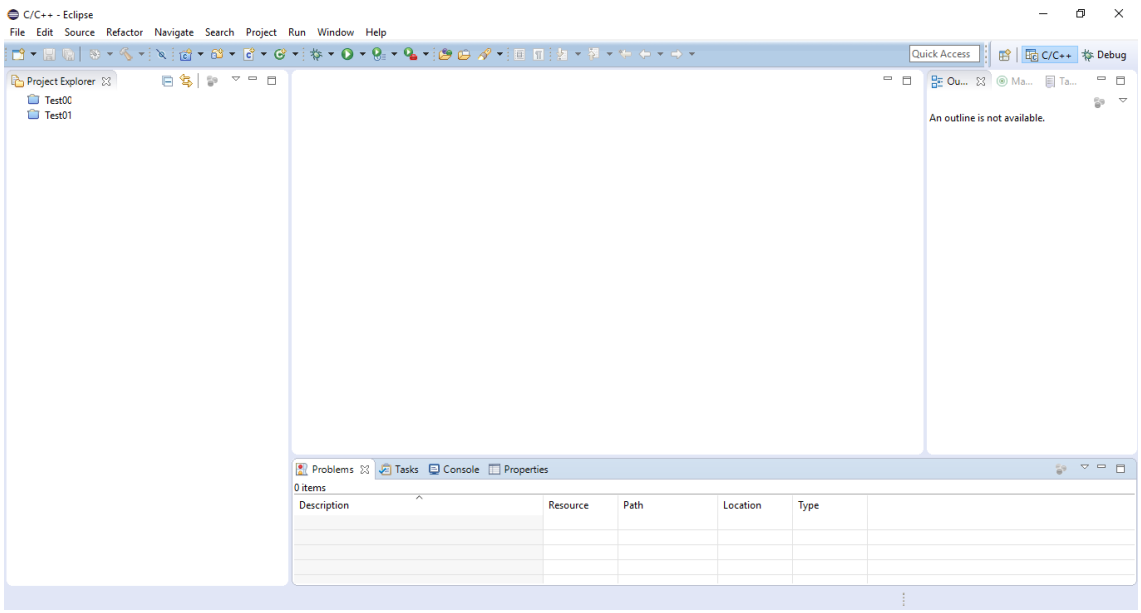


Nos da la posibilidad de elegir la ubicación del workspace del SW4STM32 que será la carpeta dónde se ubiquen los proyectos con los que se trabajan.

Yo lo dejé en la localización que viene por defecto.



Clic en ok y se abre el IDE que se observa es muy similar a cualquier IDE de eclipse (y de hecho se basa en él).

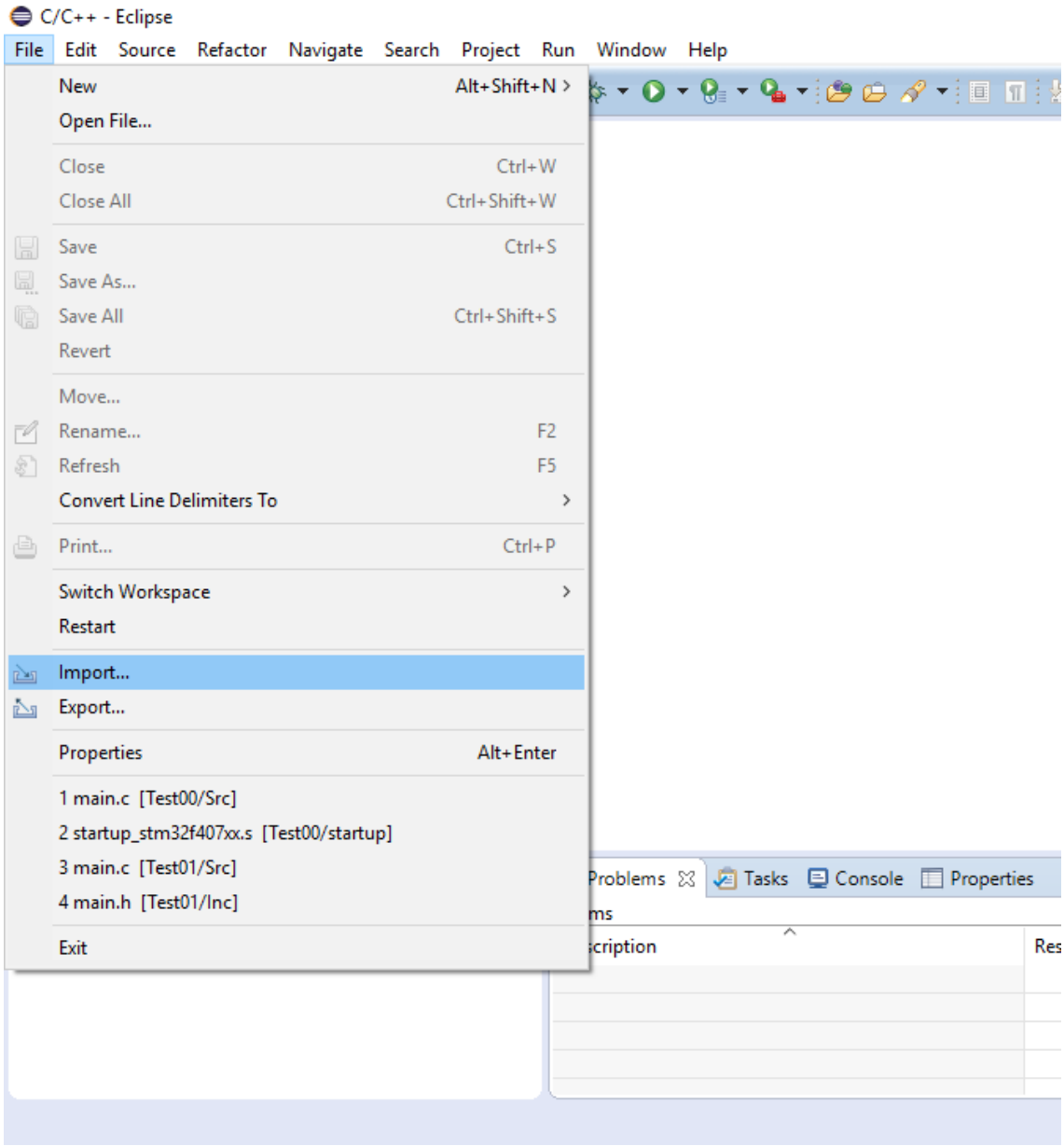


Una vez descargado y descomprimido el archivo de mi GitHub en el lugar que usted desee

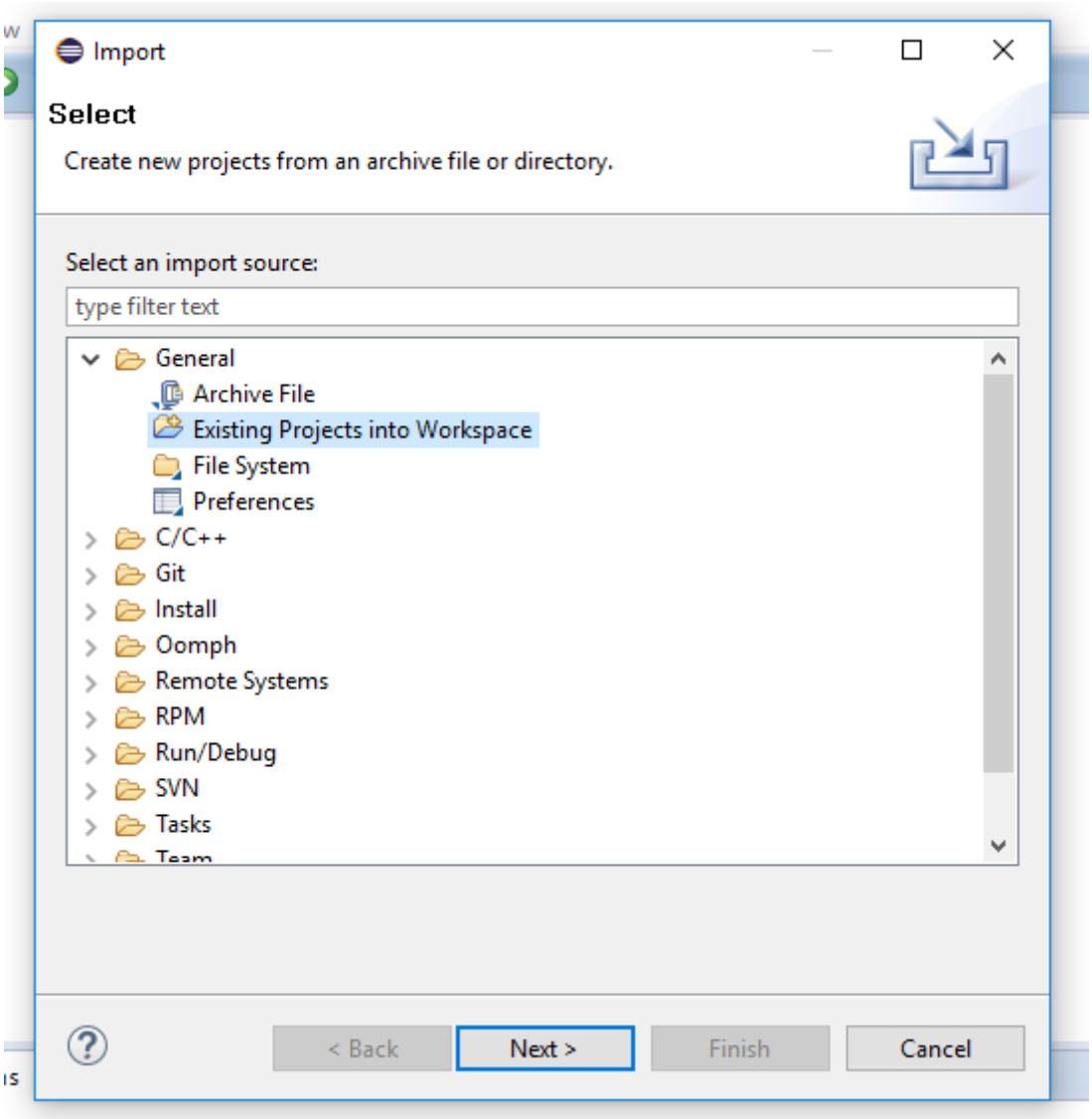
https://github.com/ecm163

procedemos a importarlo a el Workspace del IDE.

Para ello clic en la pestaña ‘File’ y clic en ‘Import…’:

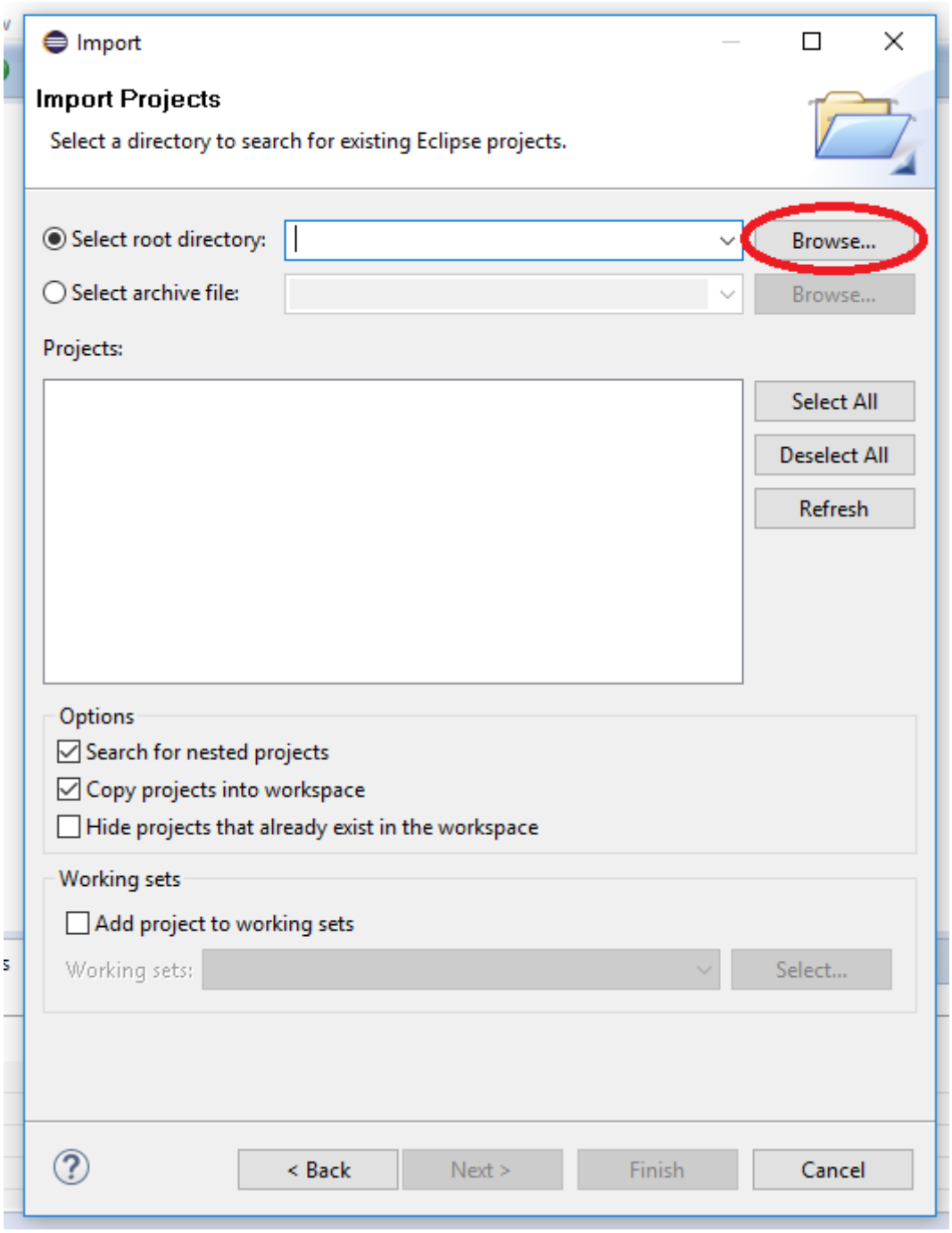


Se abre un cuadro de diálogo y en él desplegamos el directorio ‘General’ y ahí seleccionamos ‘Existing Projects into Workspace’.



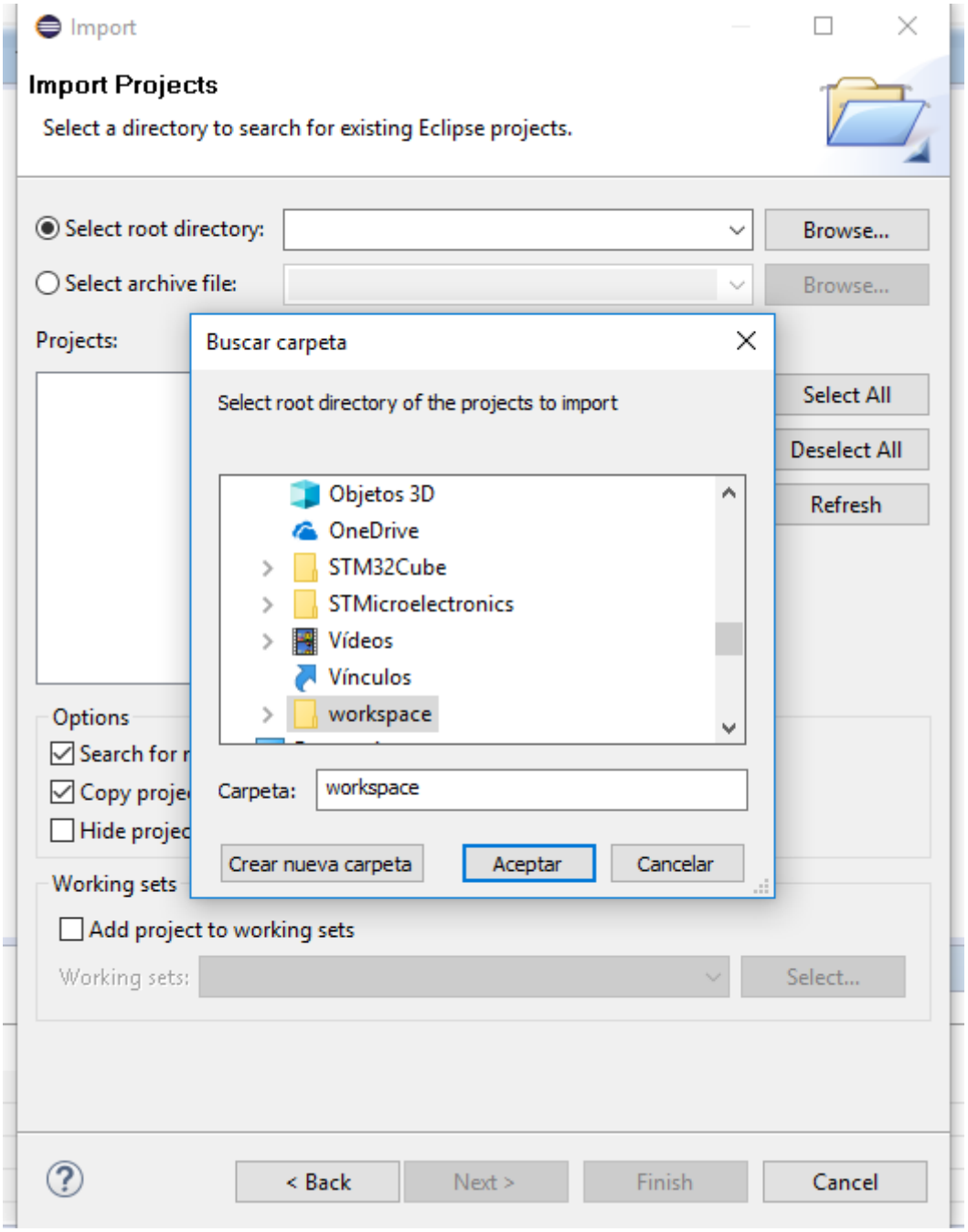
Clic en next.

Nos aparece la siguiente ventana:

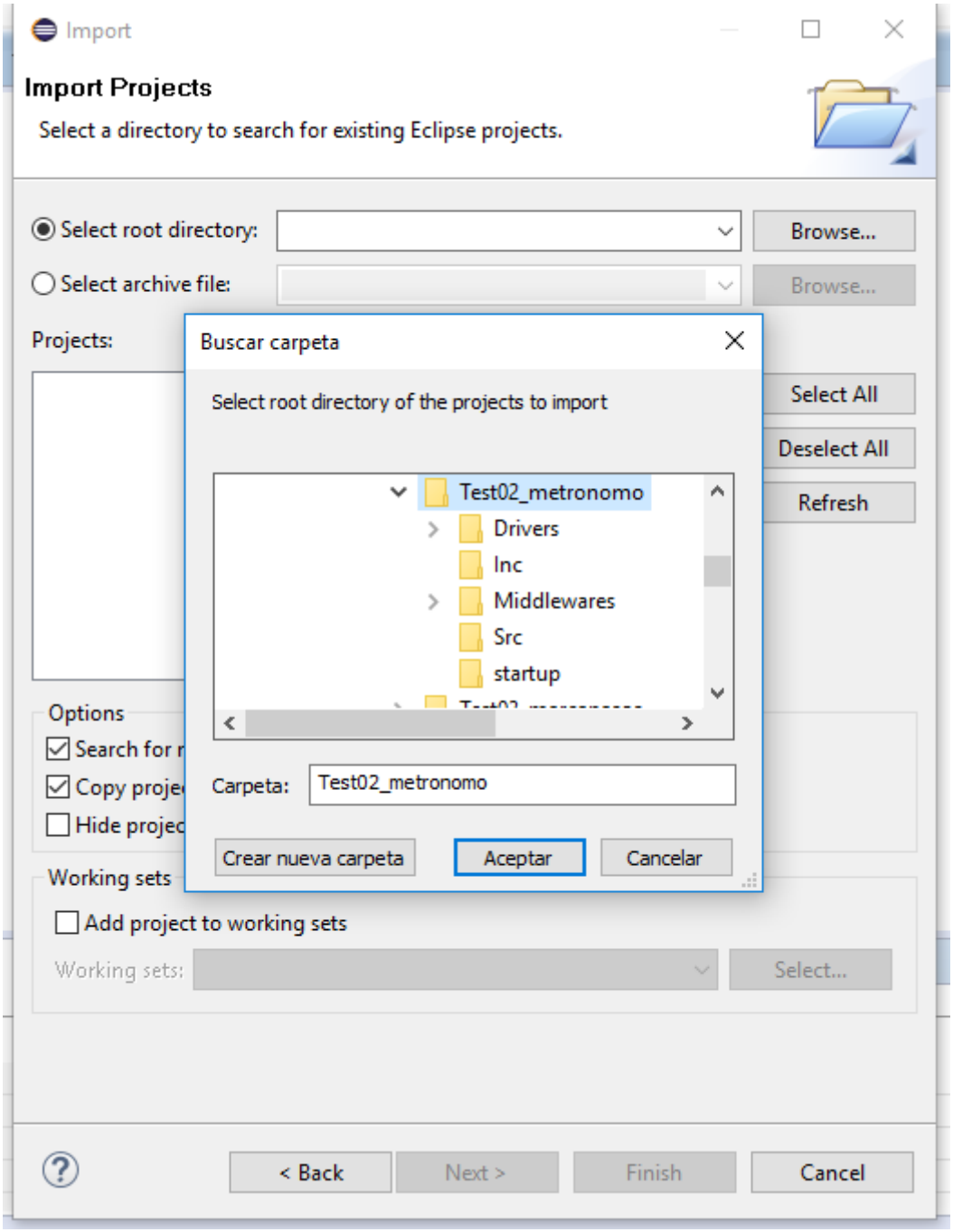


En dicha ventana clic en el botón ‘Browse’ señalado en la anterior figura.

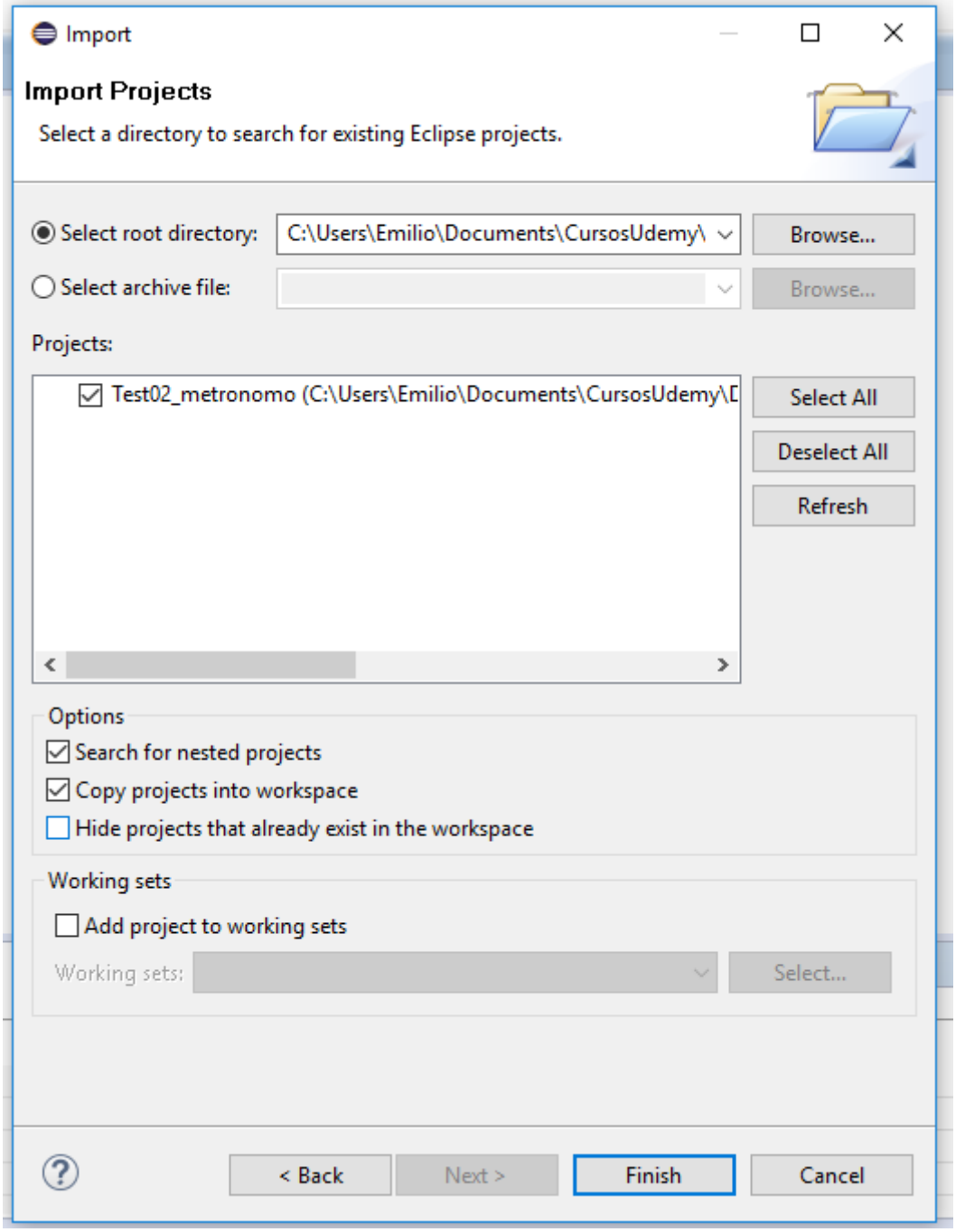
Nos aparece una nueva ventana emergente:



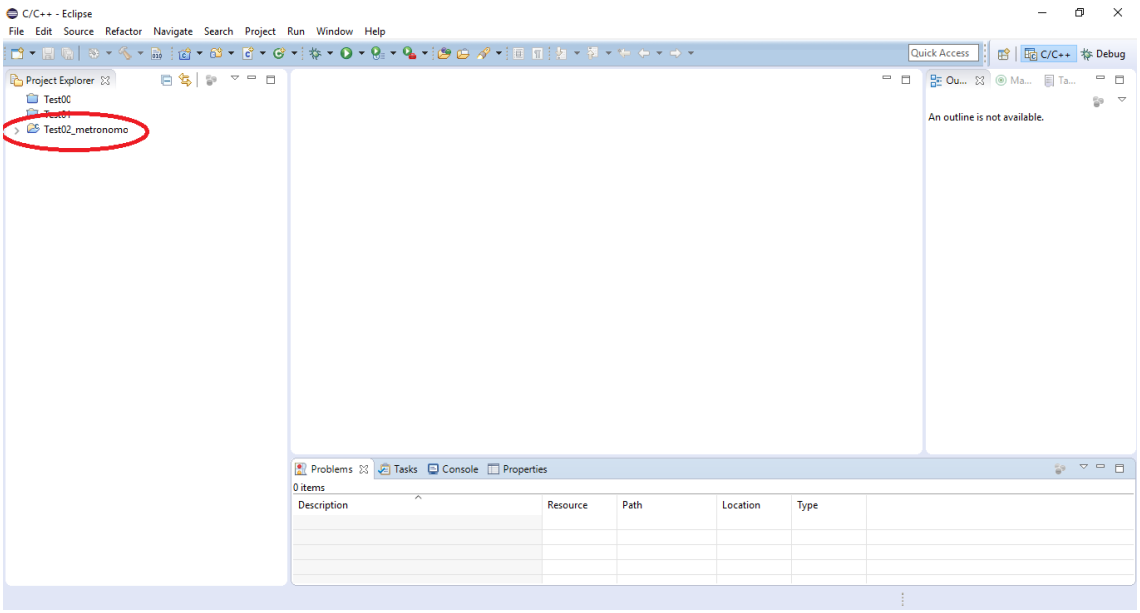
En dicha ventana navegamos hasta la ubicación del el archivo descargado y descomprimido desde mi GitHub.

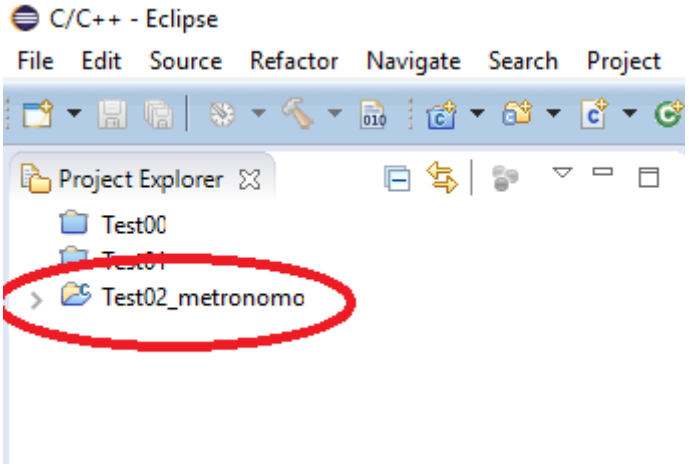


Clic en aceptar y clic en finish.

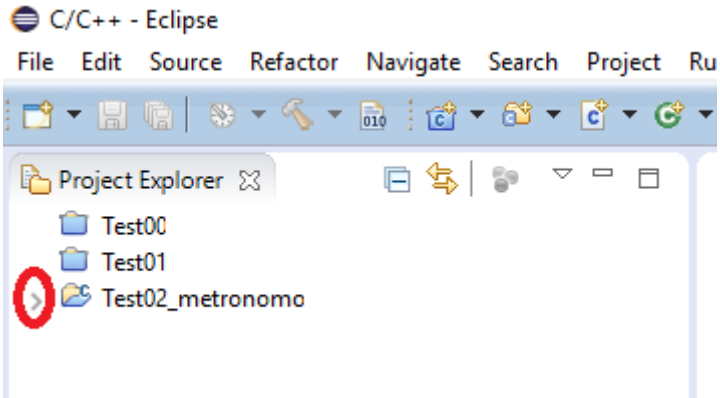


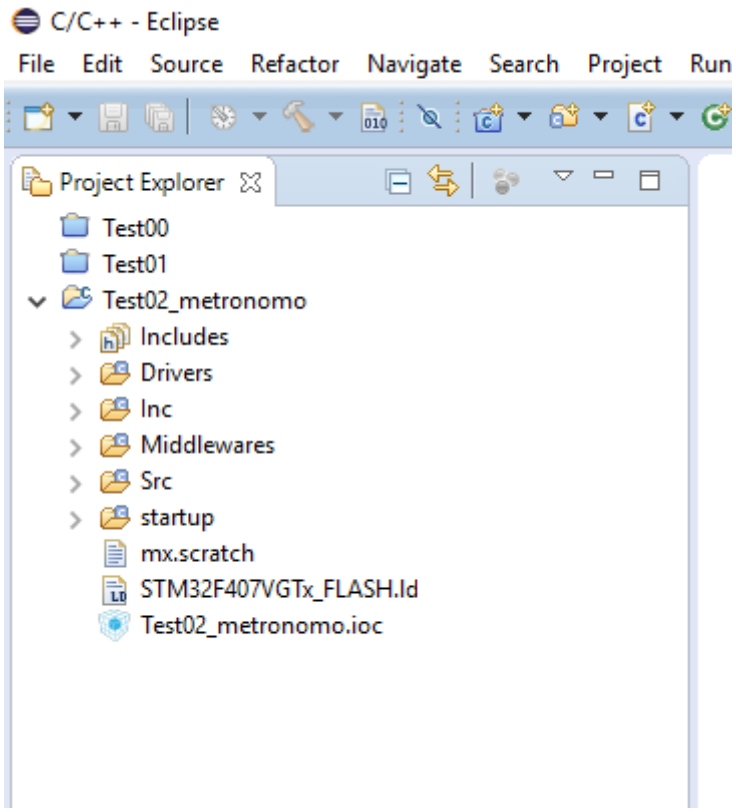
Ahora observamos que tenemos el proyecto en el panel de la izquierda “Project Explorer”.





Haciendo clic en el símbolo '>' a la izquierda de su carpeta podemos desplegar los archivos y carpetas que contiene.





• ¿Qué hace éste programa cuando se carga en la placa?.

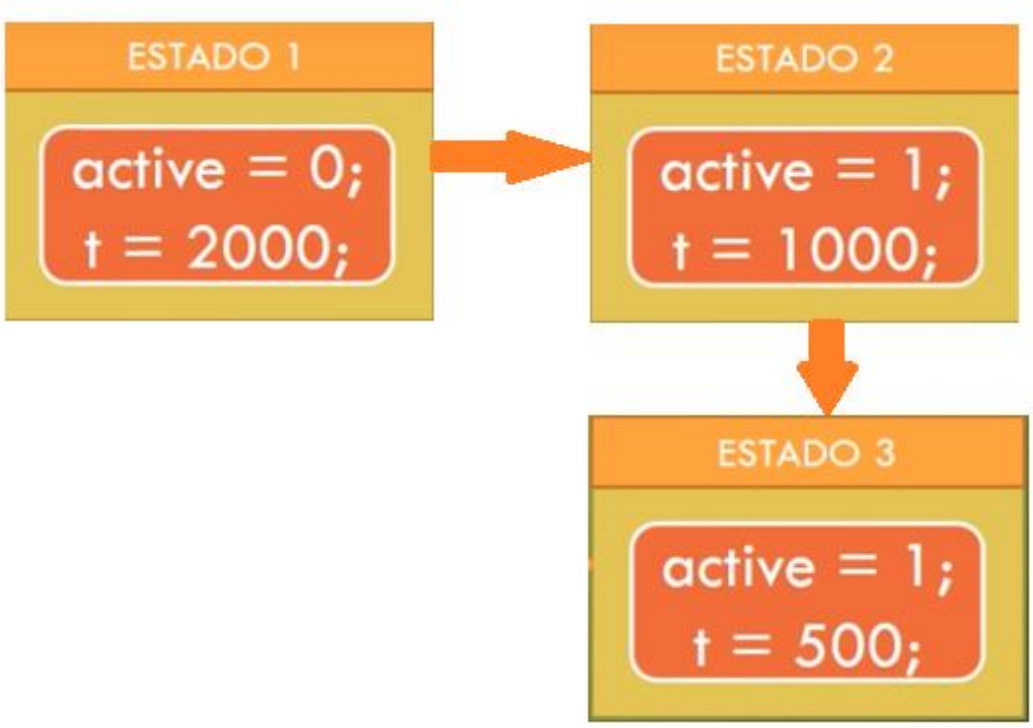
Una vez cargado el programa en la placa, si pulsamos el botón azul una primera vez, de la placa (el botón 'user'), el led naranja de dicha placa empezará a parpadear de manera que se mantiene encendido un segundo y luego se apaga un segundo, se vuelve a encender un segundo y apagar un segundo, y así sucesivamente.

Si volvemos a pulsar el botón azul o 'user' de la placa por segunda vez, ahora el parpadeo disminuye a la mitad su frecuencia, esto es, se enciende medio segundo, se paga medio segundo, se enciende medio segundo, se apaga medio segundo … y así sucesivamente.

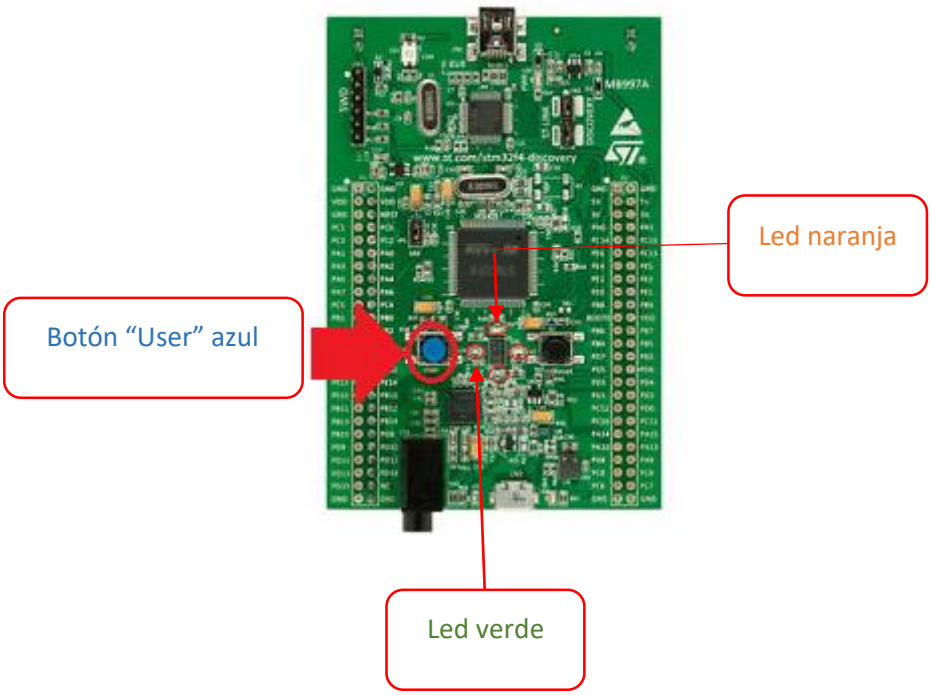
Si volvemos a pulsar el botón azul o 'user' una tercera vez el parpadeo se detiene y se inicializa el sistema para que la próxima vez que se pulse el botón tengamos el parpadeo de un segundo encendido, un segundo apagado, un segundo encendido, un segundo apagado (el mismo caso que la primera vez que se pulsa el botón).

El led verde se mantendrá en todo momento encendido para mostrar que el programa se está ejecutando.

Se podría resumir con el siguiente esquema:

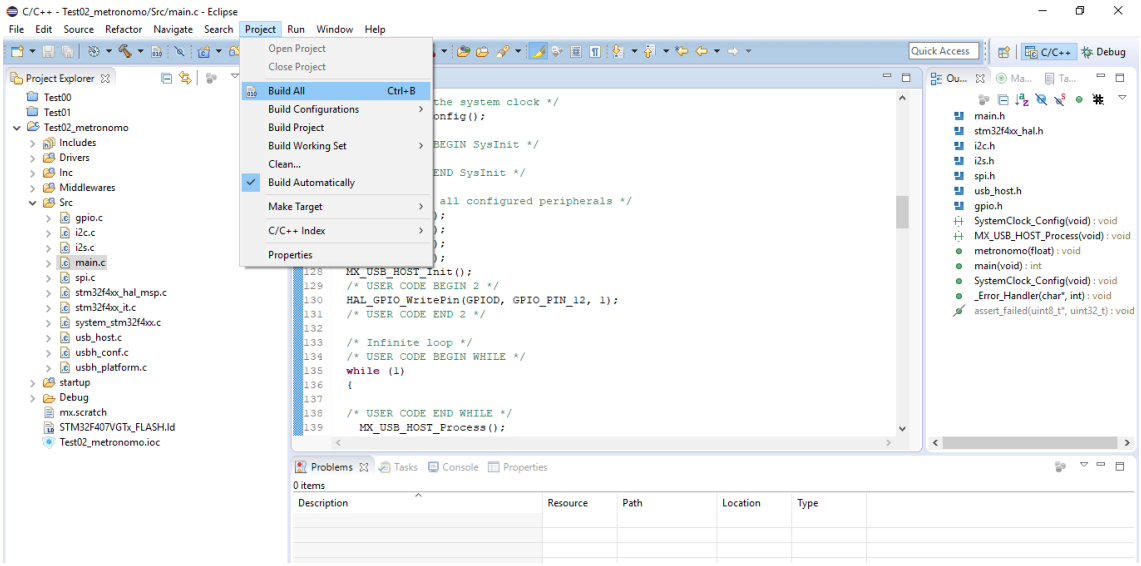


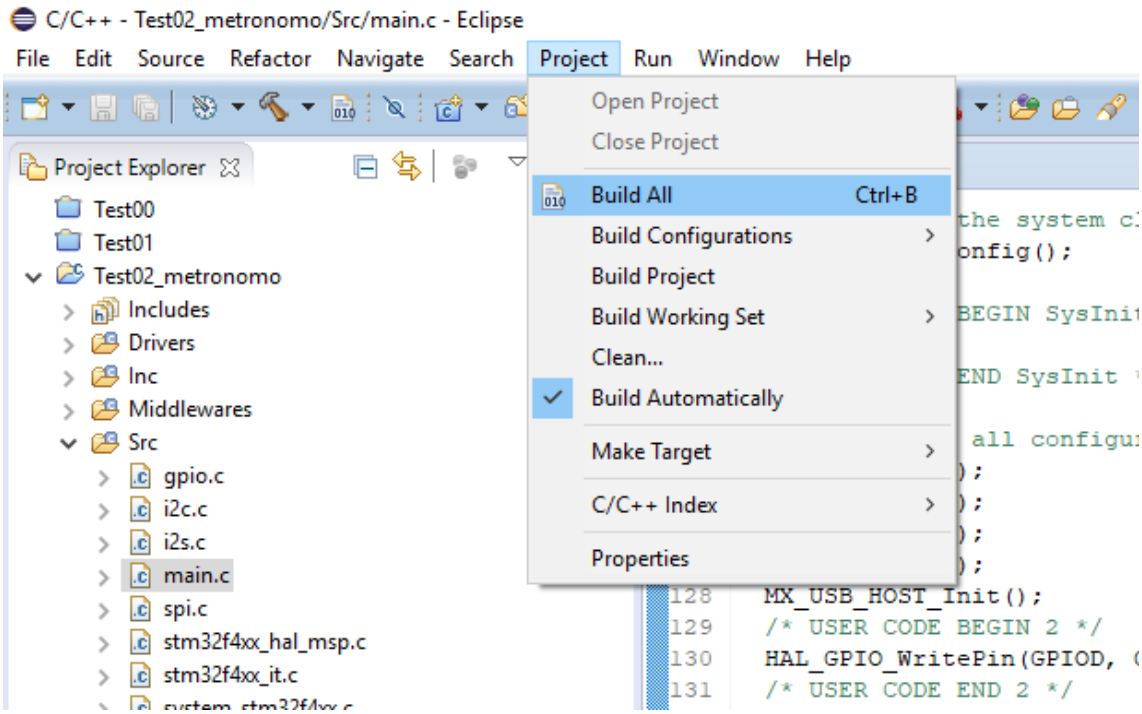
Dónde t = 2000 es t = 2000 ms = 2 sg, t = 1000 es t = 1000 ms = 1 sg y t = 500 es t = 500 ms = 0,5 sg representan el tiempo de encendido o apagado del LED naranja.



• Carga y ejecución del programa en la placa.

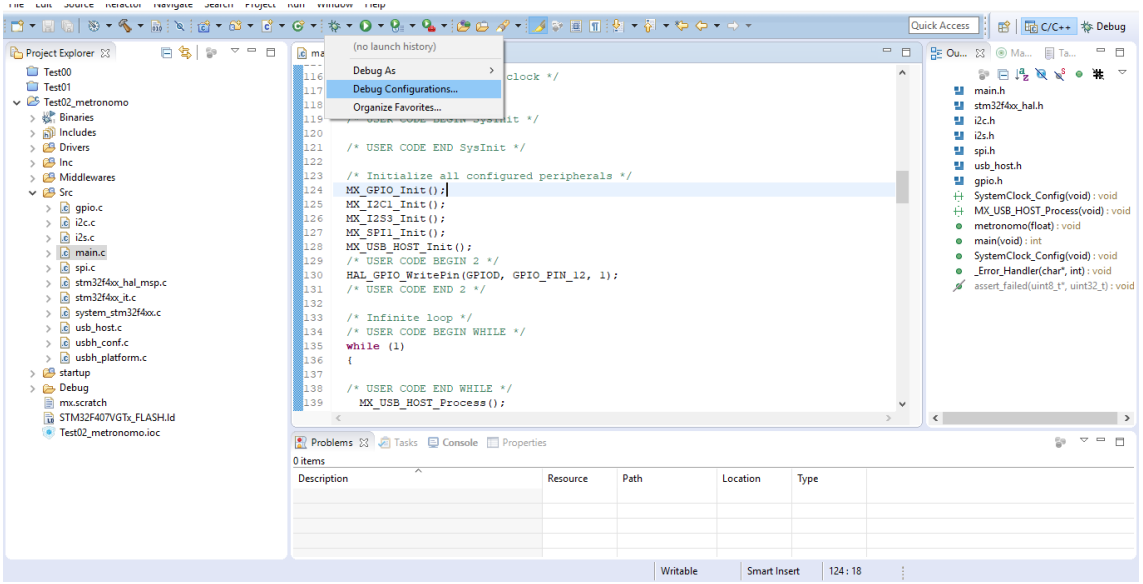
Una vez que tenemos el programa listo, lo compilamos. Para ello hacer clic en ‘Project’ y en el desplegable hacer clic en ‘Buid all’.

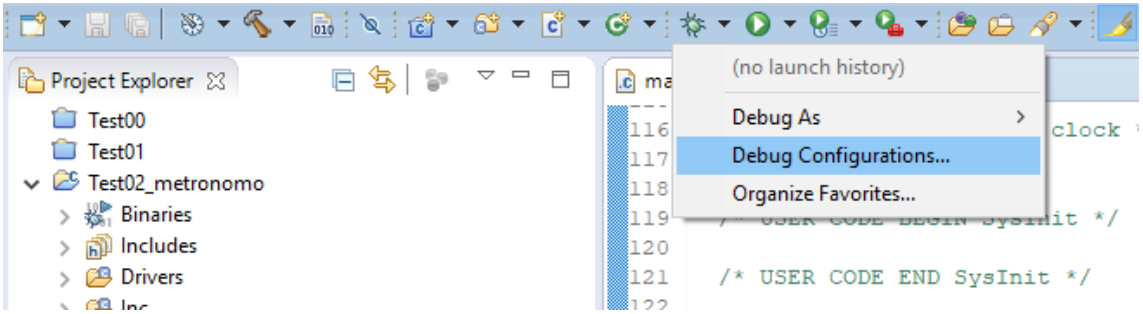




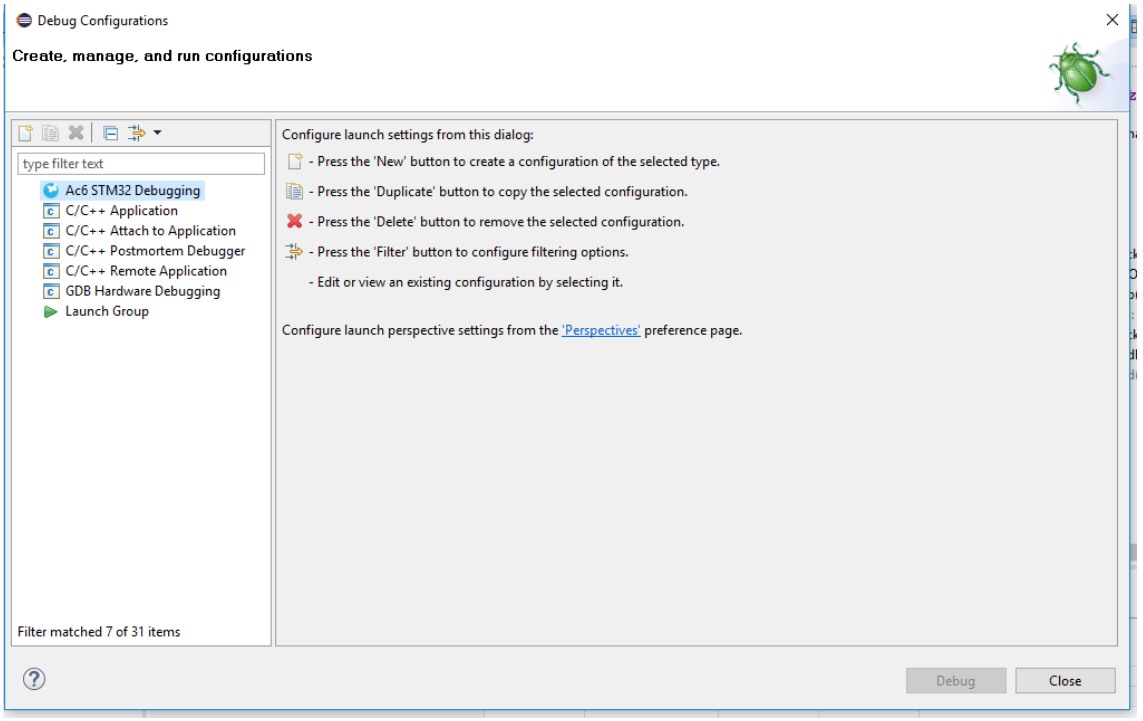
Esto generará el archivo que usará el micro para ejecutar el programa.

Una vez que hemos acabado de compilar el proyecto abrimos el ‘Debug Configuration’ como sigue:

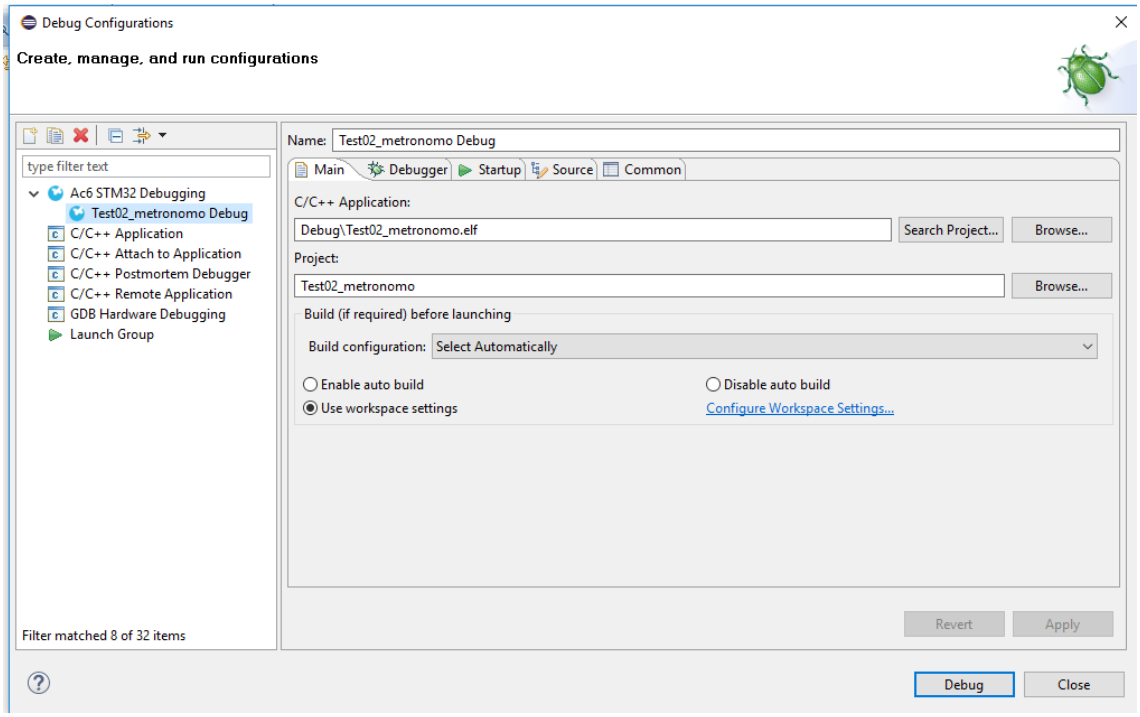




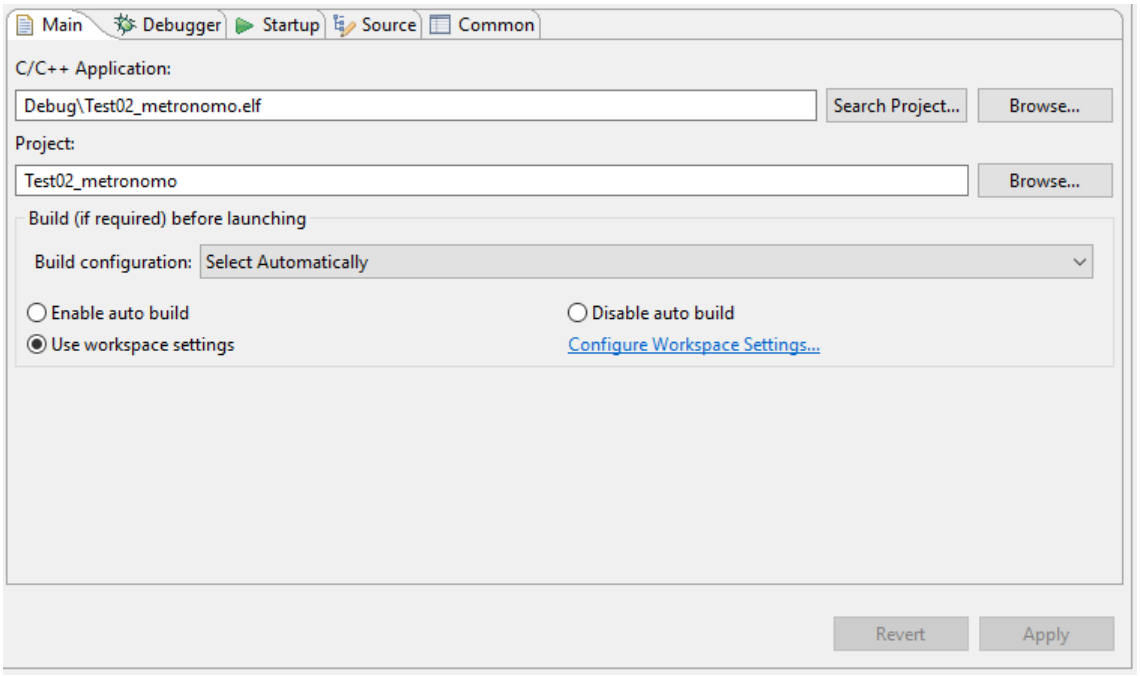
Aparece la ventana:

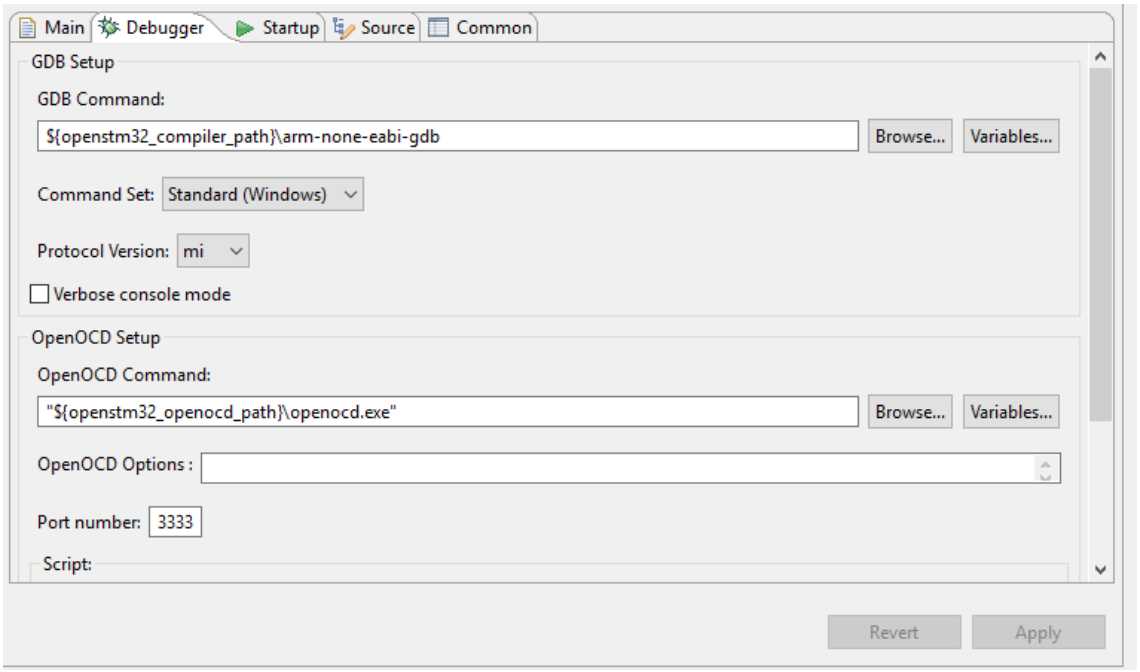


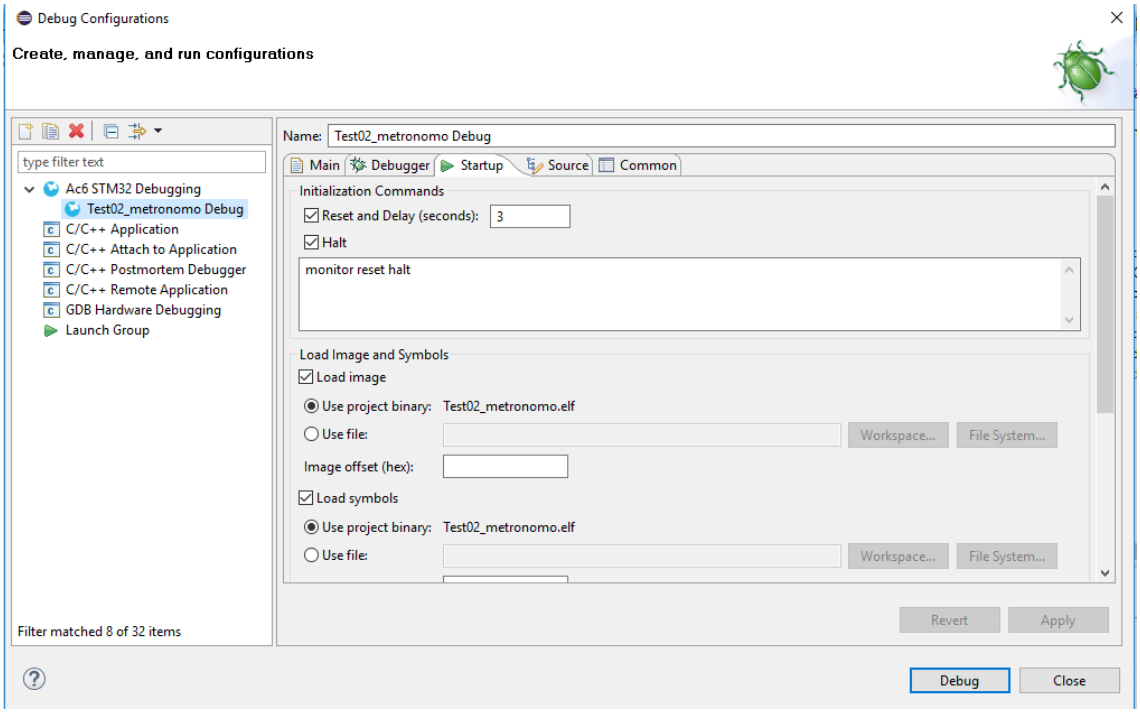
Hacer doble clic en ‘Ac6 STM32 Debugging’.

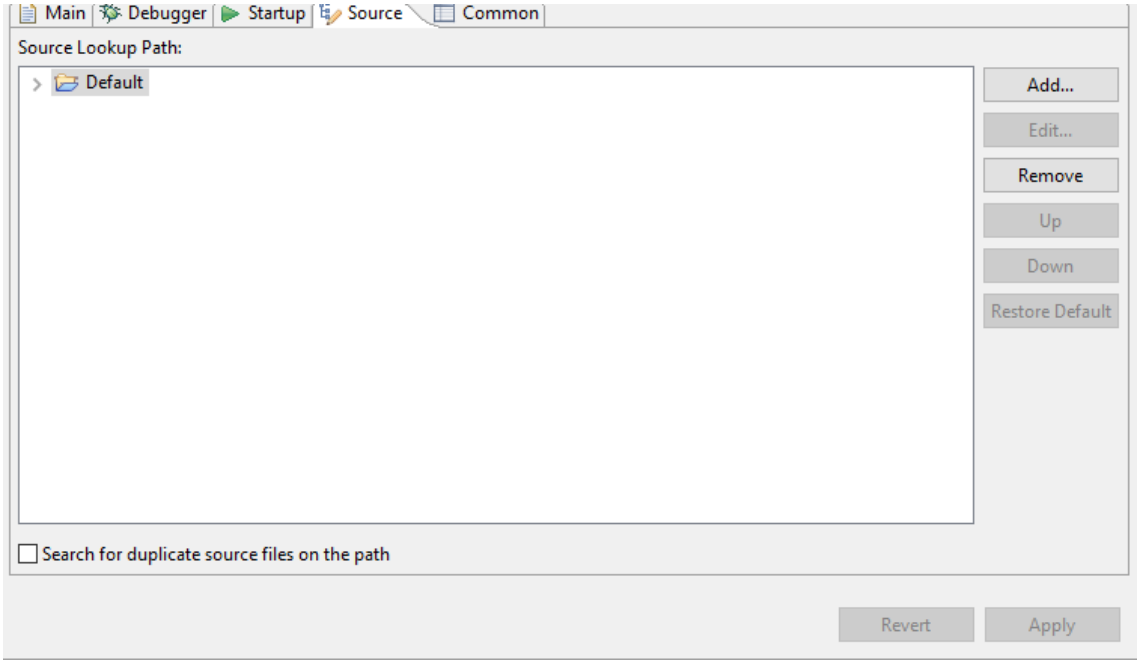


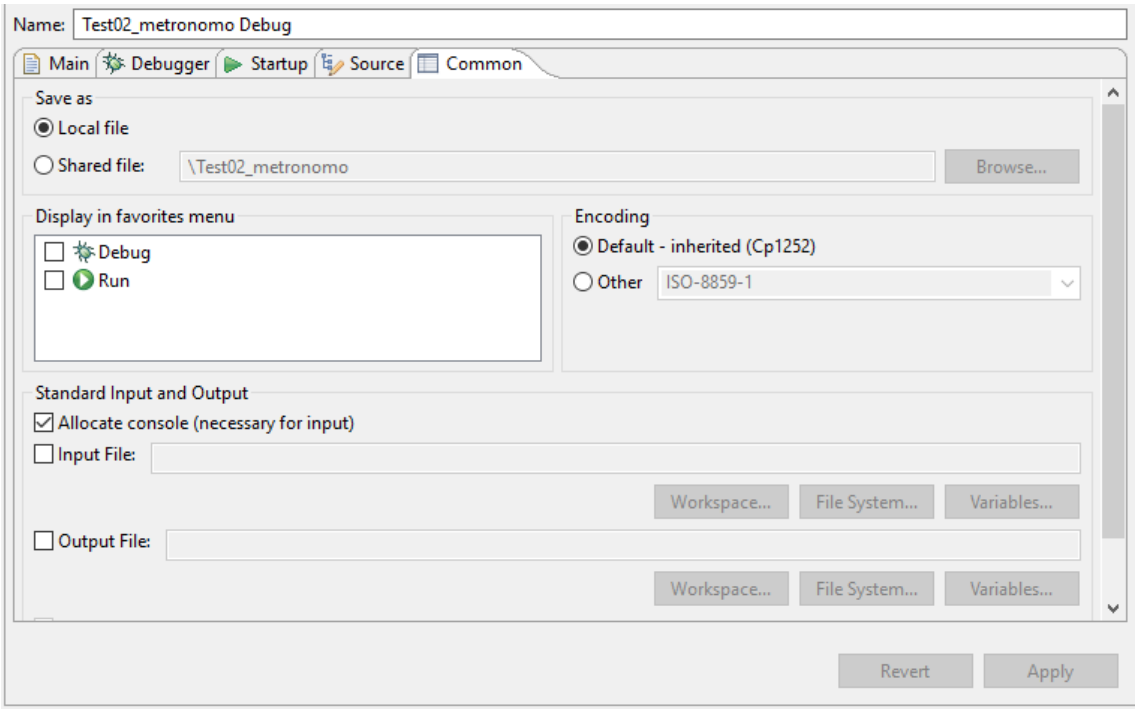
Configurar las distintas pestañas como sigue (que normalmente será la configuración por defecto):



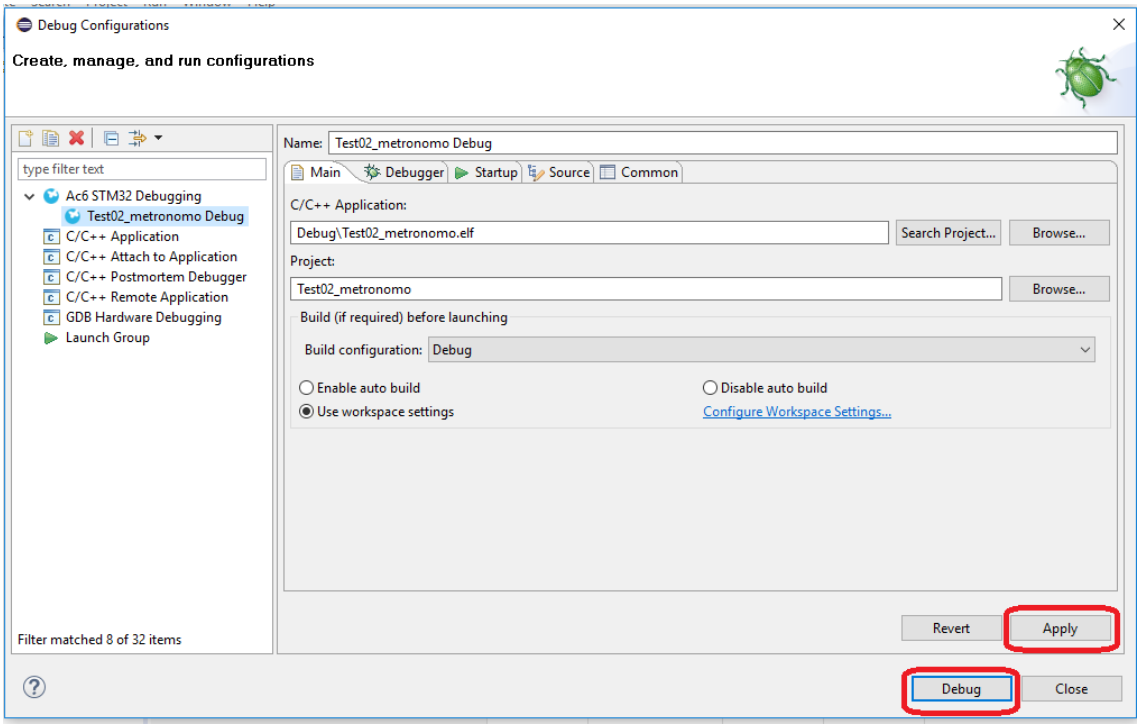

Main
Debugger
Startup
Source
Common
GDB Setup
GDB Command:
${openstm32_compiler_path}\arm-none-eabi-gdb
Browse...
Variables...
Command Set:
Standard (Windows)
Protocol Version:
mi
Verbose console mode
OpenOCD Setup
OpenOCD Command:
"${openstm32_openocd_path}\openocd.exe"
Browse...
Variables...
OpenOCD Options :
Port number:
3333
Script:
Revert
Apply


Debug Configurations
Create, manage, and run configurations
type filter text
Ac6 STM32 Debugging
Test02_metronomo Debug
C/C++ Application
C/C++ Attach to Application
C/C++ Postmortem Debugger
C/C++ Remote Application
GDB Hardware Debugging
Launch Group
Filter matched 8 of 32 items
Name:
Test02_metronomo Debug
Main
Debugger
Startup
Source
Common
Initialization Commands
Reset and Delay (seconds):
3
Halt
monitor reset halt
Load Image and Symbols
Load image
Use project binary:
Test02_metronomo.elf
Use file:
Workspace...
File System...
Image offset (hex):
Load symbols
Use project binary:
Test02_metronomo.elf
Use file:
Workspace...
File System...
Revert
Apply
Debug
Close

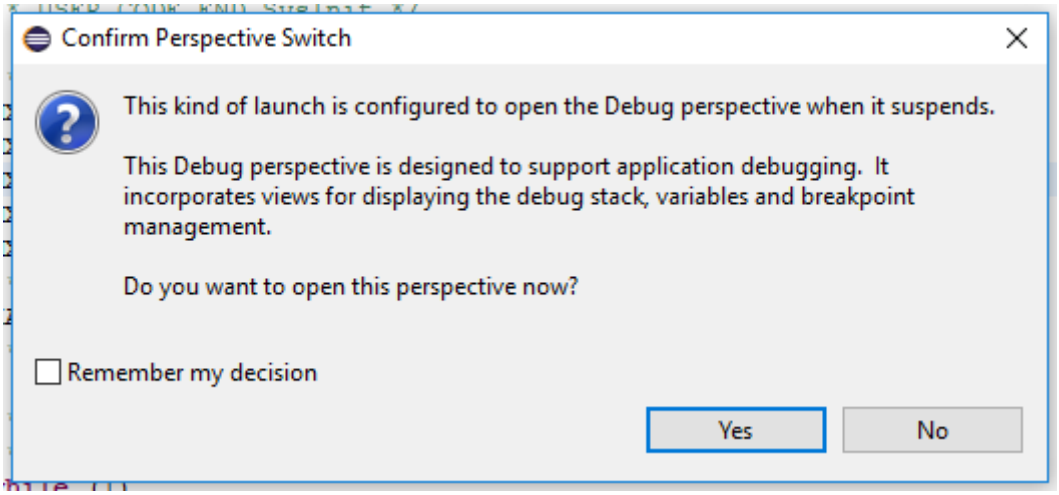




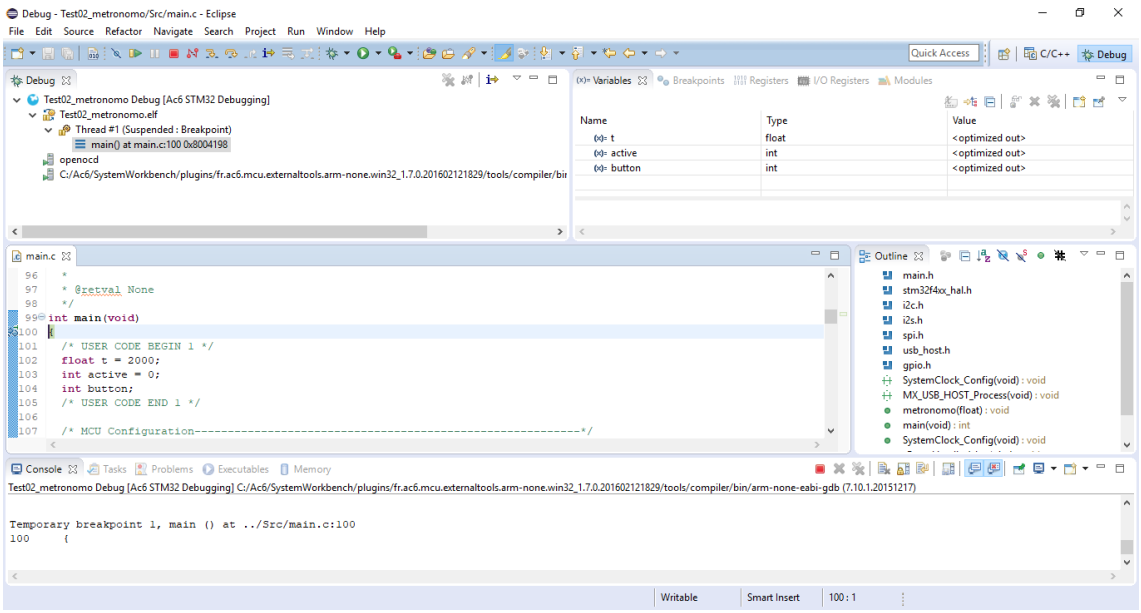
A continuación, clic en ‘Apply’ y clic en ‘Debug’.



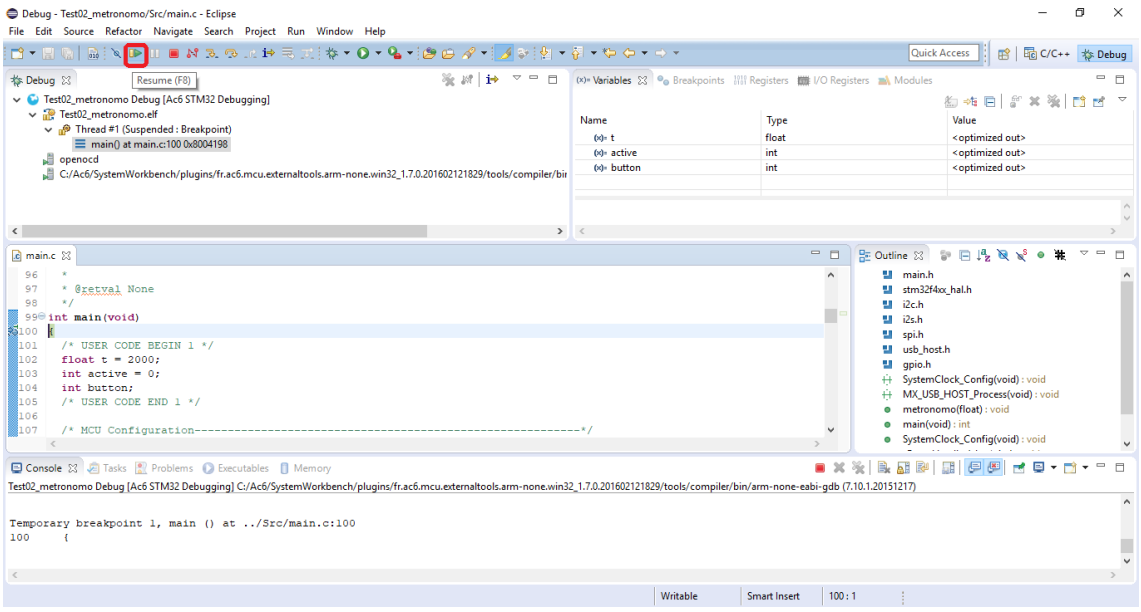
Nos aparece la siguiente ventana emergente:

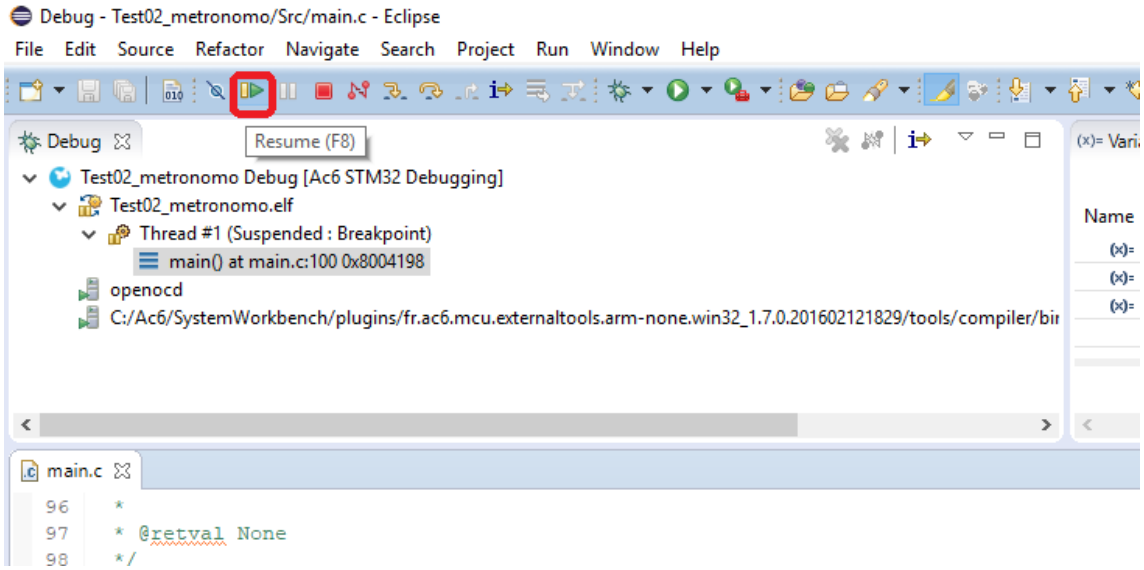


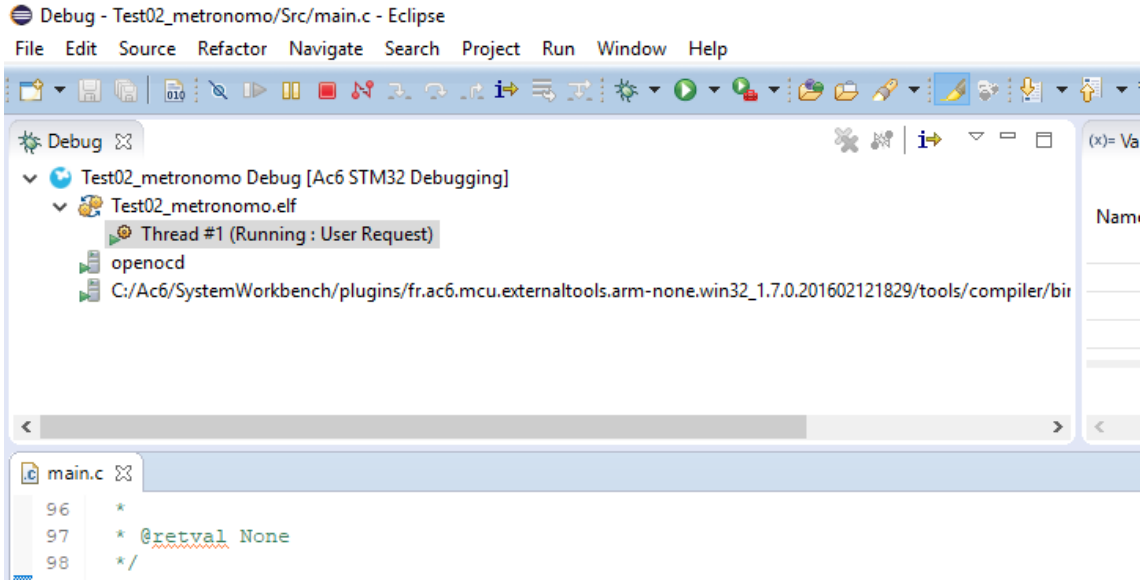
Si hacemos clic en ‘Yes’, tendremos la vista del programa en modo depuración. Hacemos clic en ‘Yes’ `y tenemos:



Ahora hacemos clic en ‘Resume’:

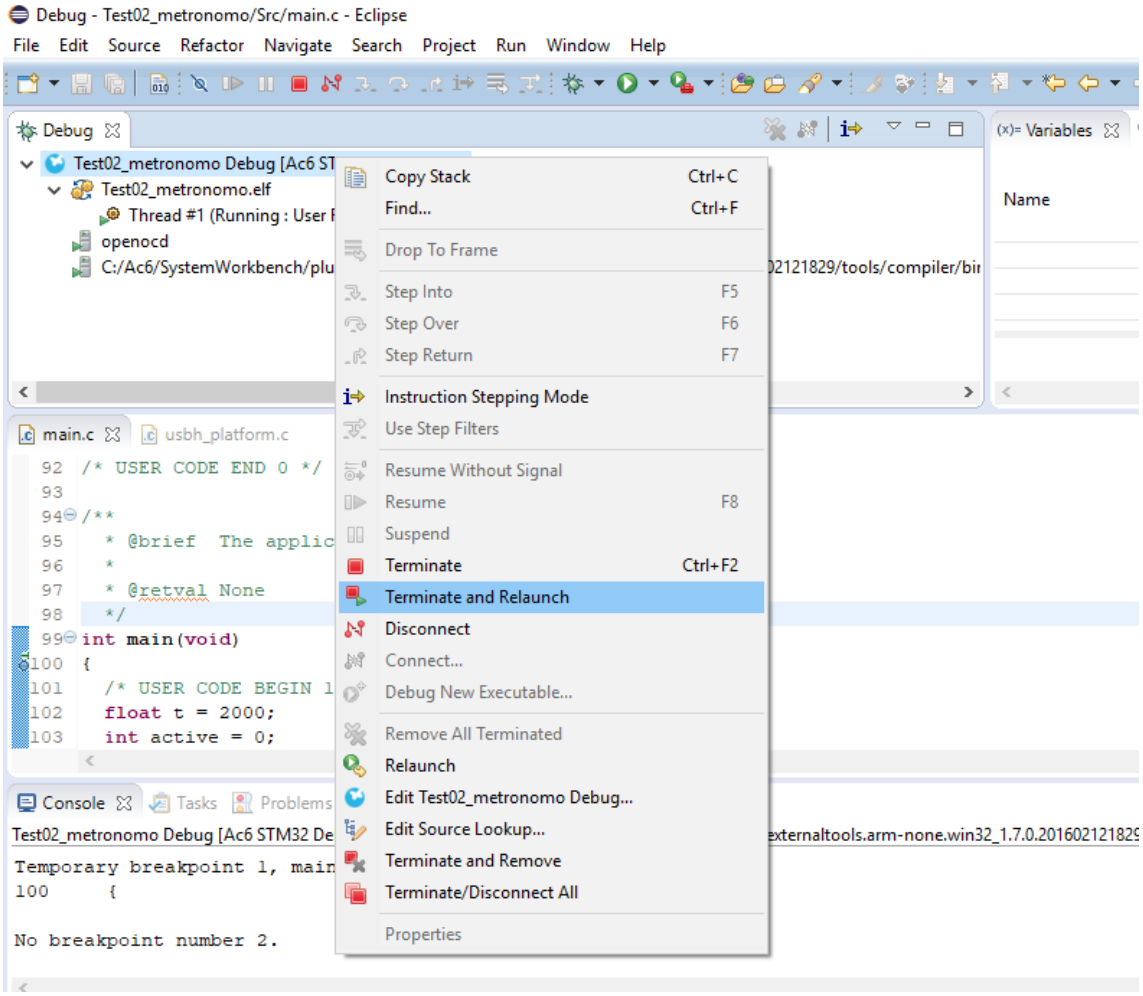






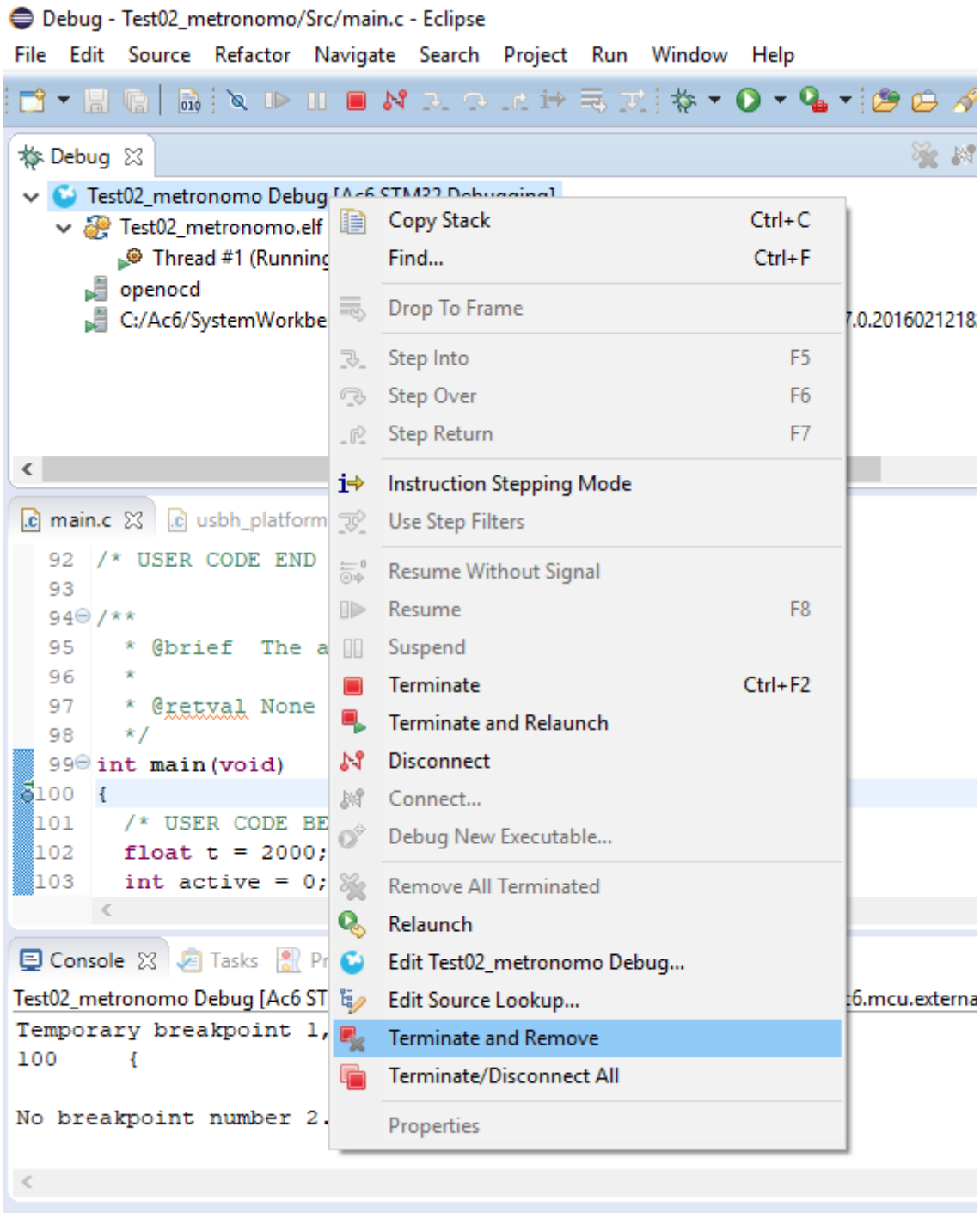
El programa ya se ha descargado en la placa y podemos pulsar el botón ‘User’ de color azul para comprobar su funcionamiento.

SI queremos reiniciar el código desde el principio, clic derecho dónde se indica en la siguiente figura, y luego “Terminate and Relaunch” en el desplegable.

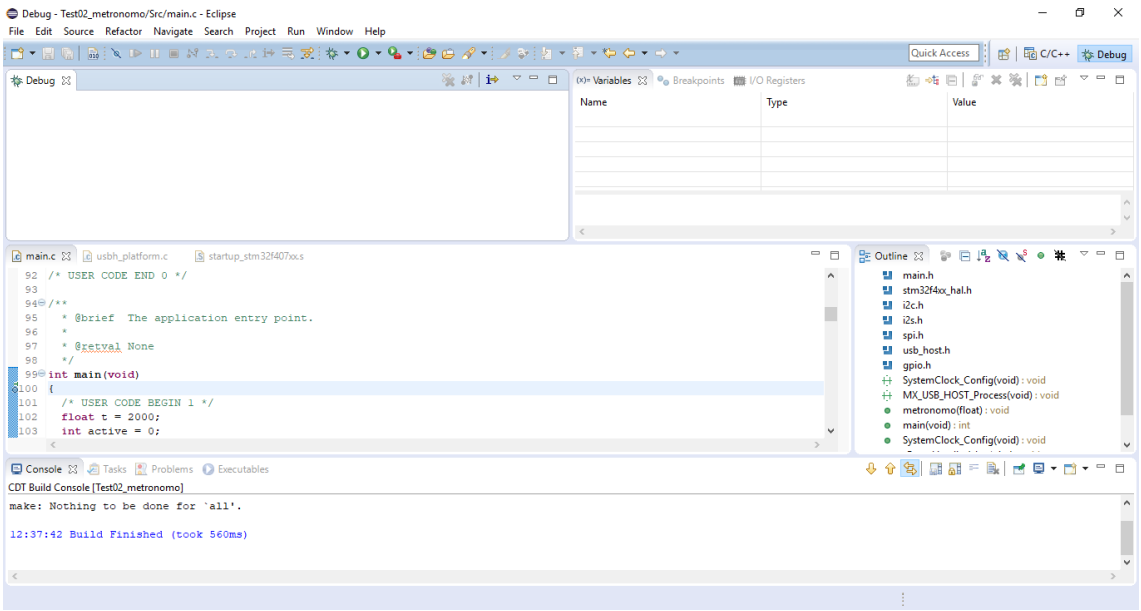


Luego podemos volver a hacer clic en ‘Resume’.

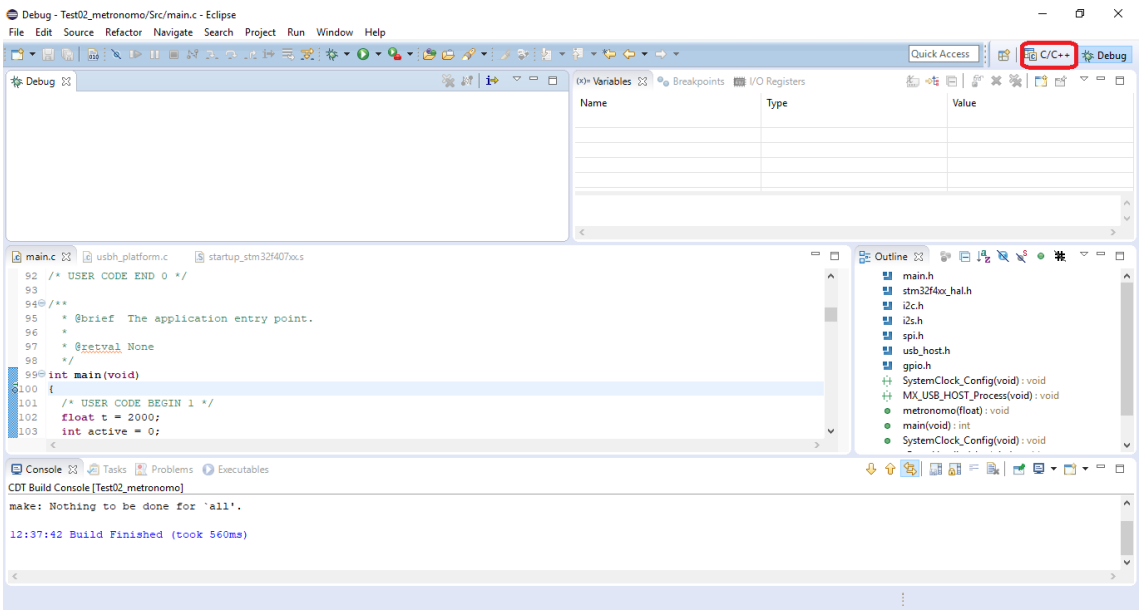
Si se desea terminar y eliminar la depuración en curso:

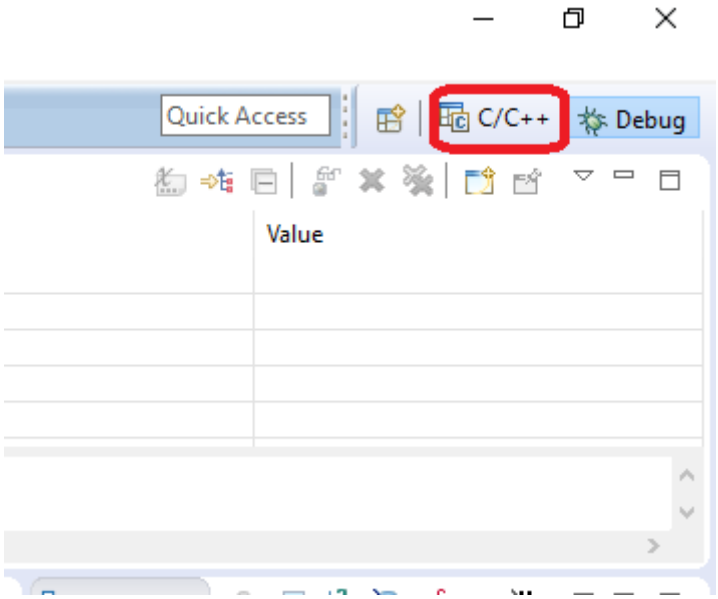


En cuyo caso tendremos:



Una vez que hemos eliminado la sesión de depuración, si queremos probar el programa de nuevo, clic en la siguiente pestaña:





Se vuelve a la visión estándar del código y ahí podemos hacer las modificaciones que creamos oportunas y volver al proceso desde la compilación tal y como se ha explicado.