

Diseño de Marcapasos Sencillo 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 μ Vision5 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 64 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](#)

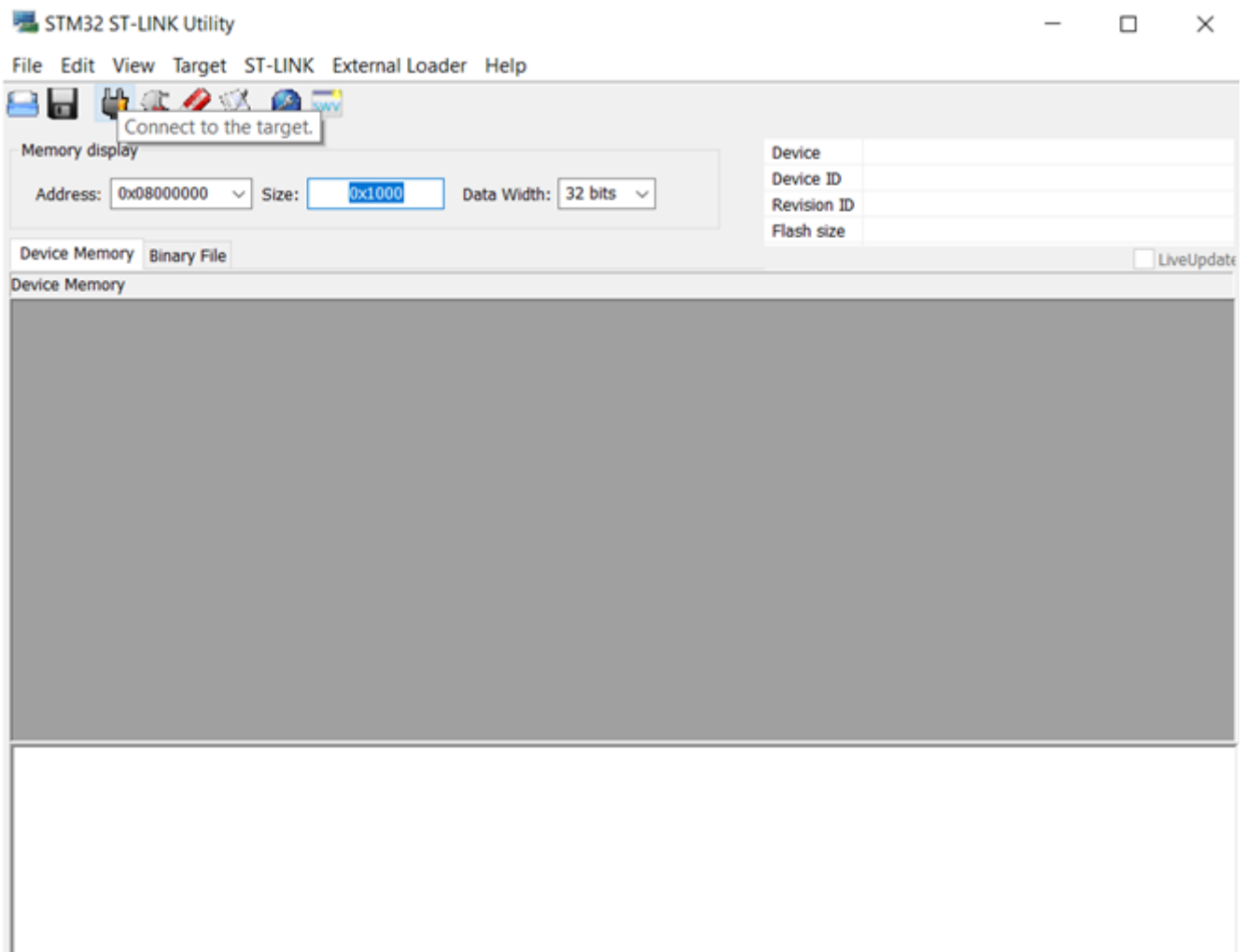
Puede descargar el archivo en la carpeta que desee y navegar a dicha carpeta para proceder a la instalación del IDE.

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

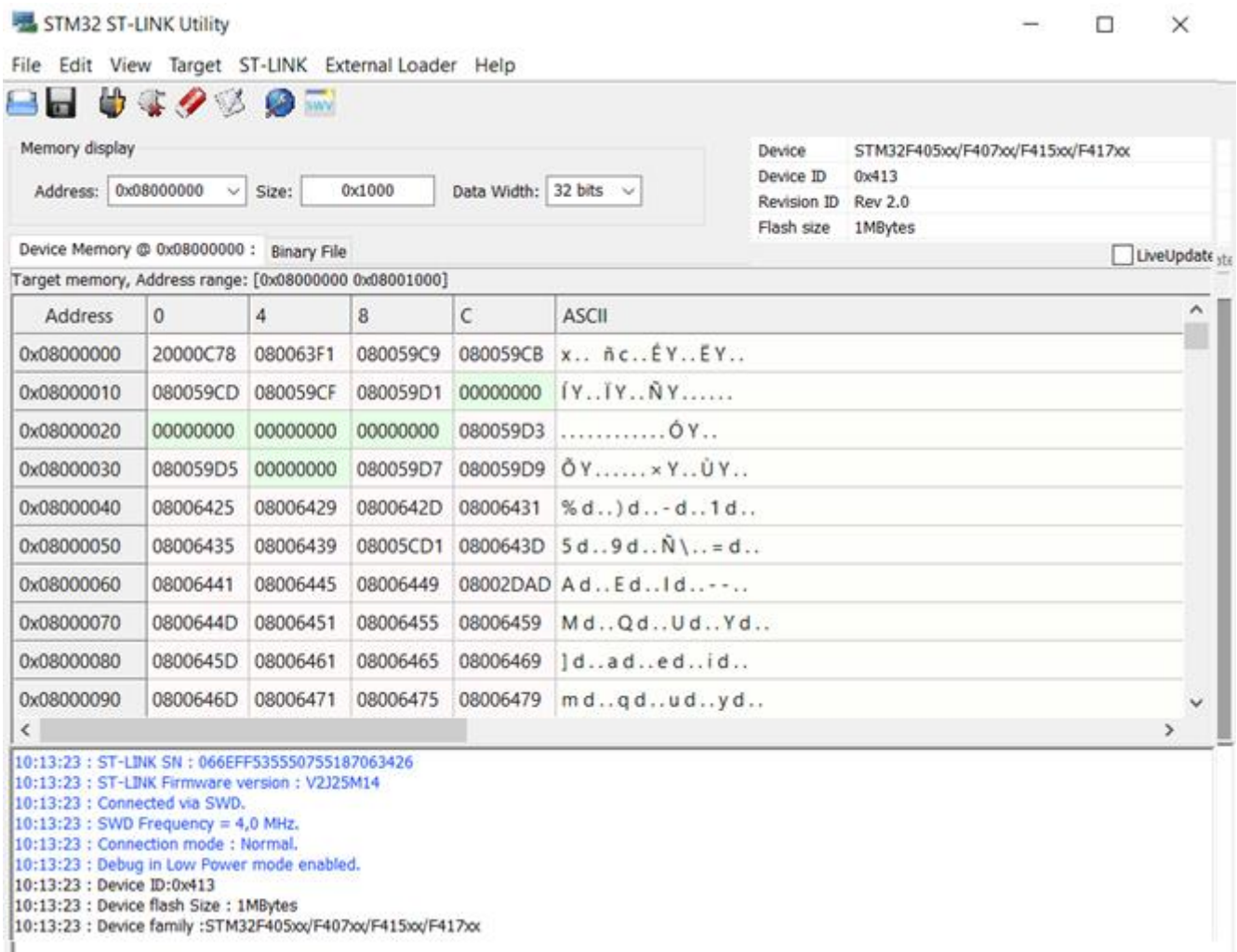
<https://www.st.com/en/development-tools/stsw-link004.html>

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:

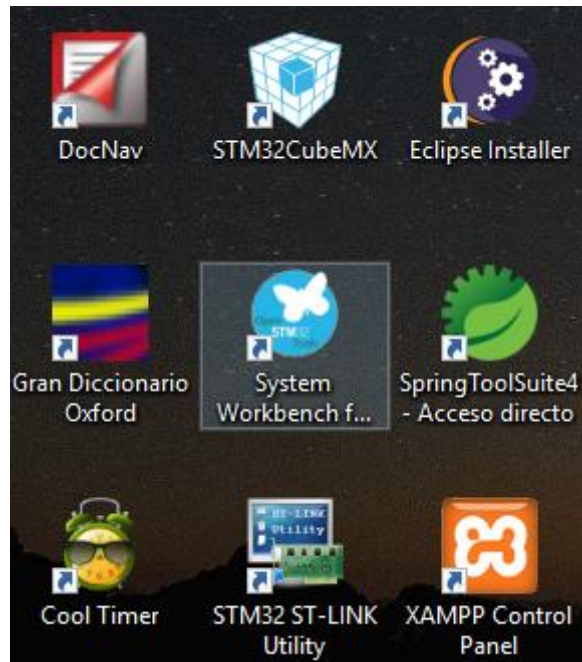
- Programar el microcontrolador (Menú Target/Program).
- Borrar el microcontrolador.
- Visualizar el estado e identificadores del microcontrolador.

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>

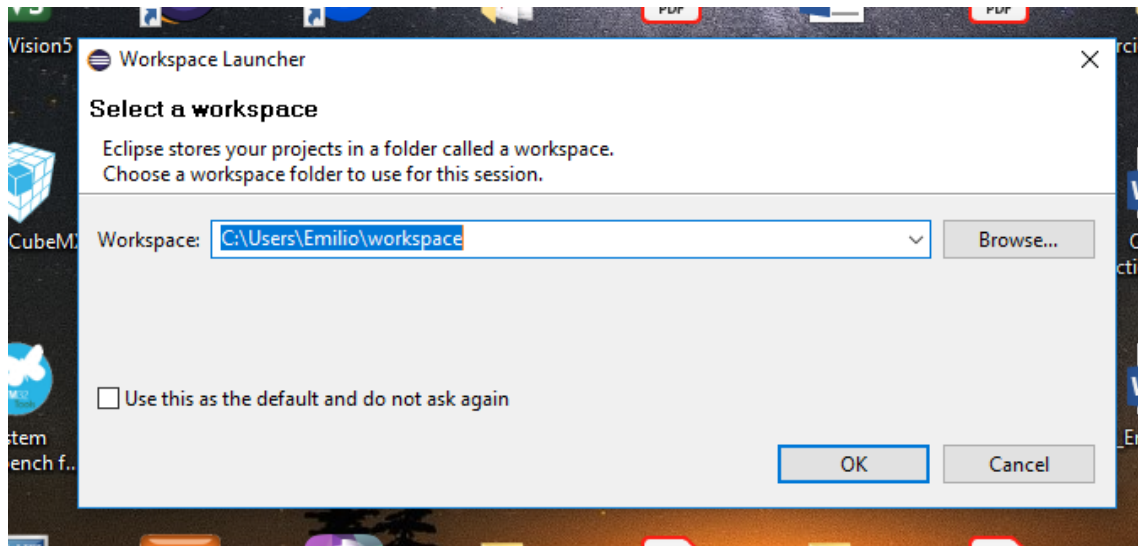
- **Abrir el programa desde el IDE SW4STM32 (System Workbench for STM32):**

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

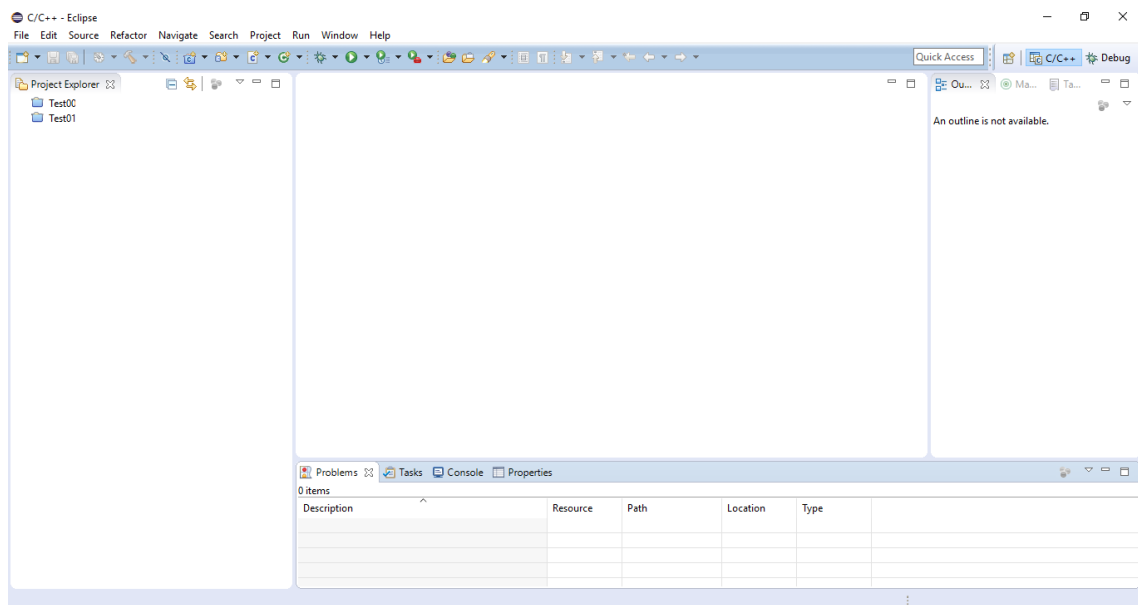


Nos da la posibilidad de elegir la ubicación del workspace del SW4STM32 que será la carpeta dónde se ubiquen los proyectos con los que se trabajan.

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).



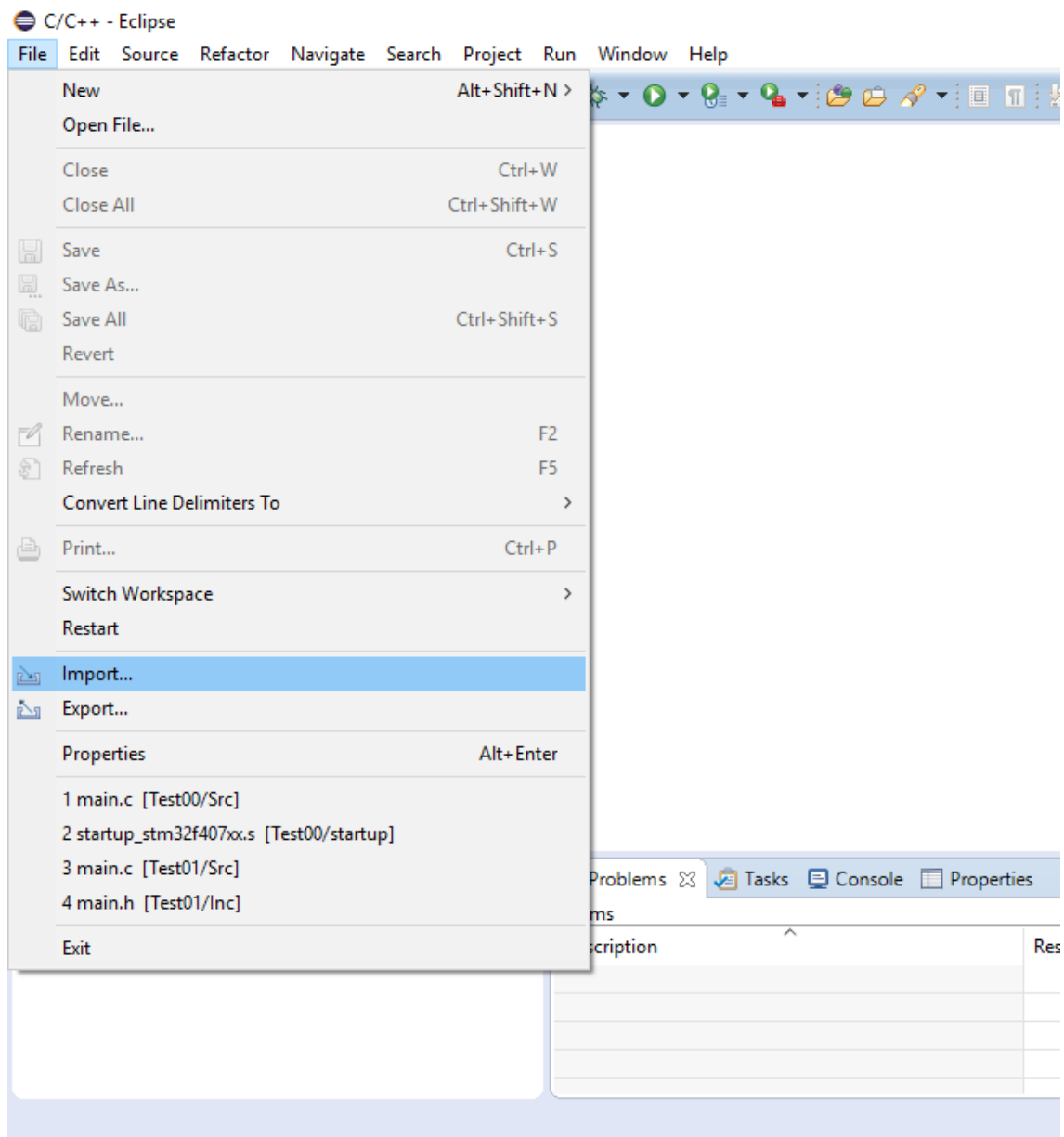
Una vez descargado y descomprimido el archivo de mi GitHub en el lugar que usted desee

<https://github.com/ecm163>

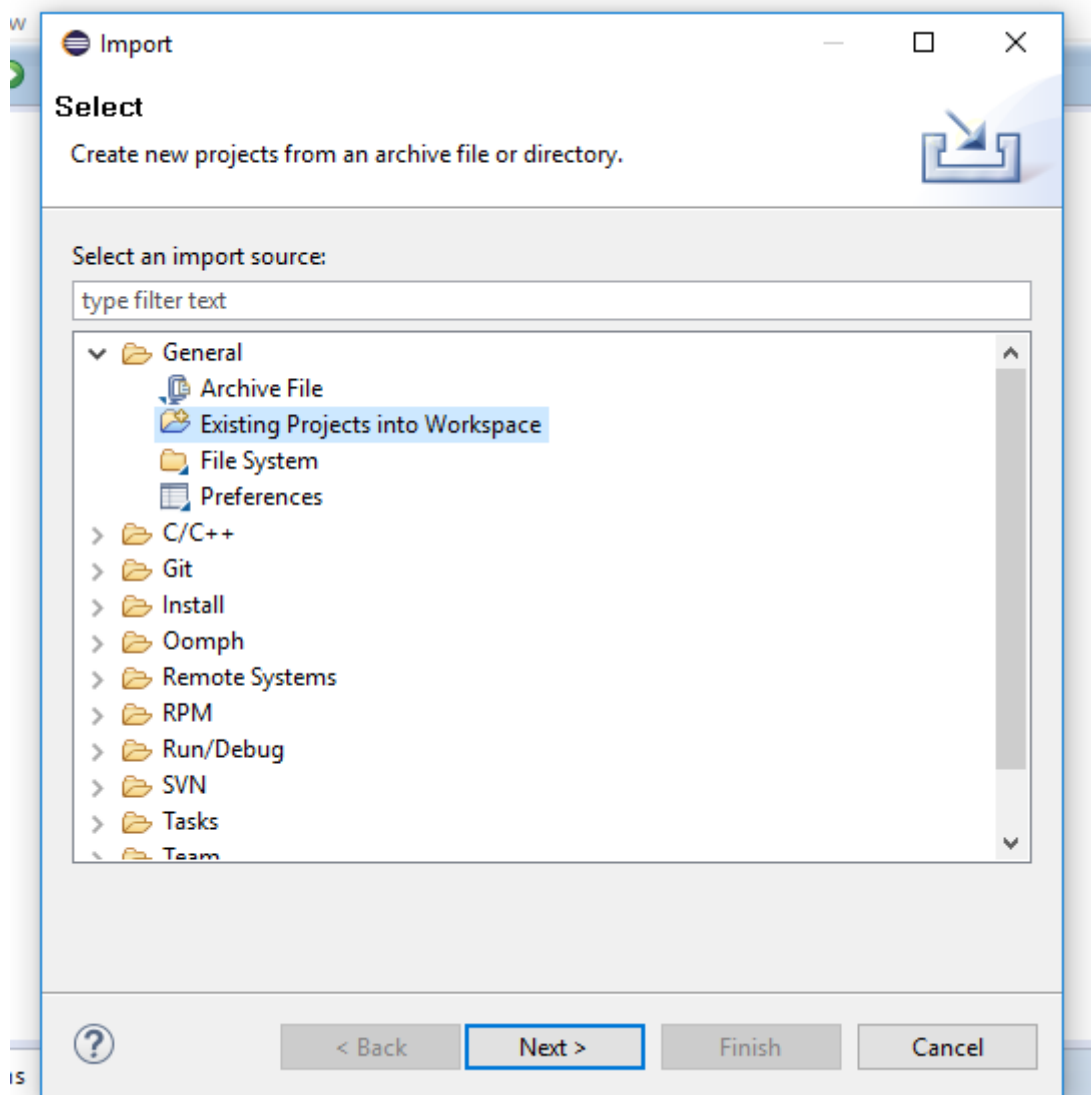
procedemos a importarlo a el Workspace del IDE.

Para ello clic en la pestaña 'File' y clic en 'Import...':

Nota Importante: el proceso es exactamente el mismo que el usado en otro proyecto subido a éste GutHub llamado 'Test02_Metronomo', por eso he usado las capturas de pantalla de ese proyecto dónde usted verá Test02_Metronomo en lugar de 'Test03_Marcapasos' (para horrar tiempo). Siga exactamente los pasos que se indica, pero con 'Test03_Marcapasos'.

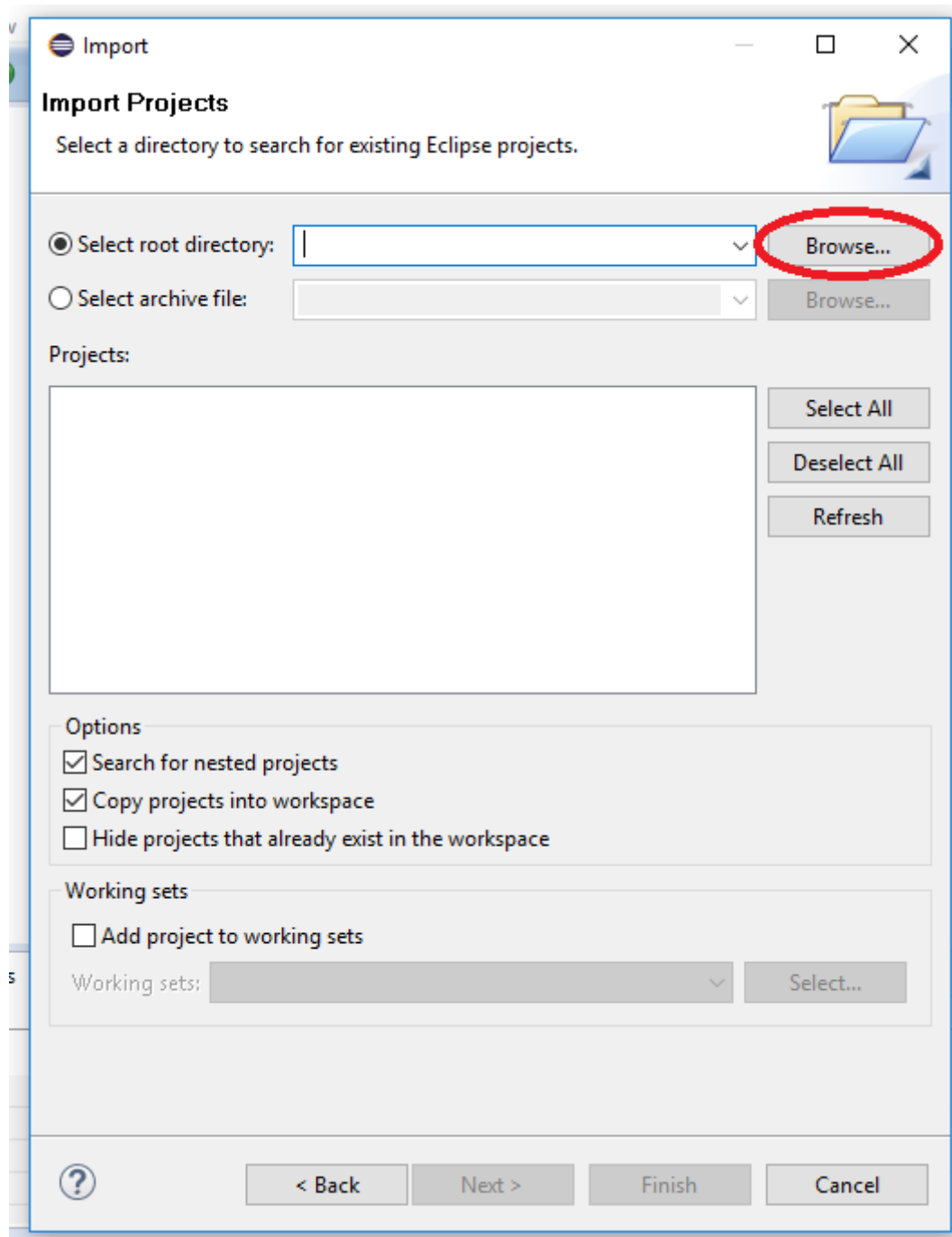


Se abre un cuadro de diálogo y en él desplegamos el directorio 'General' y ahí seleccionamos 'Existing Projects into Workspace'.



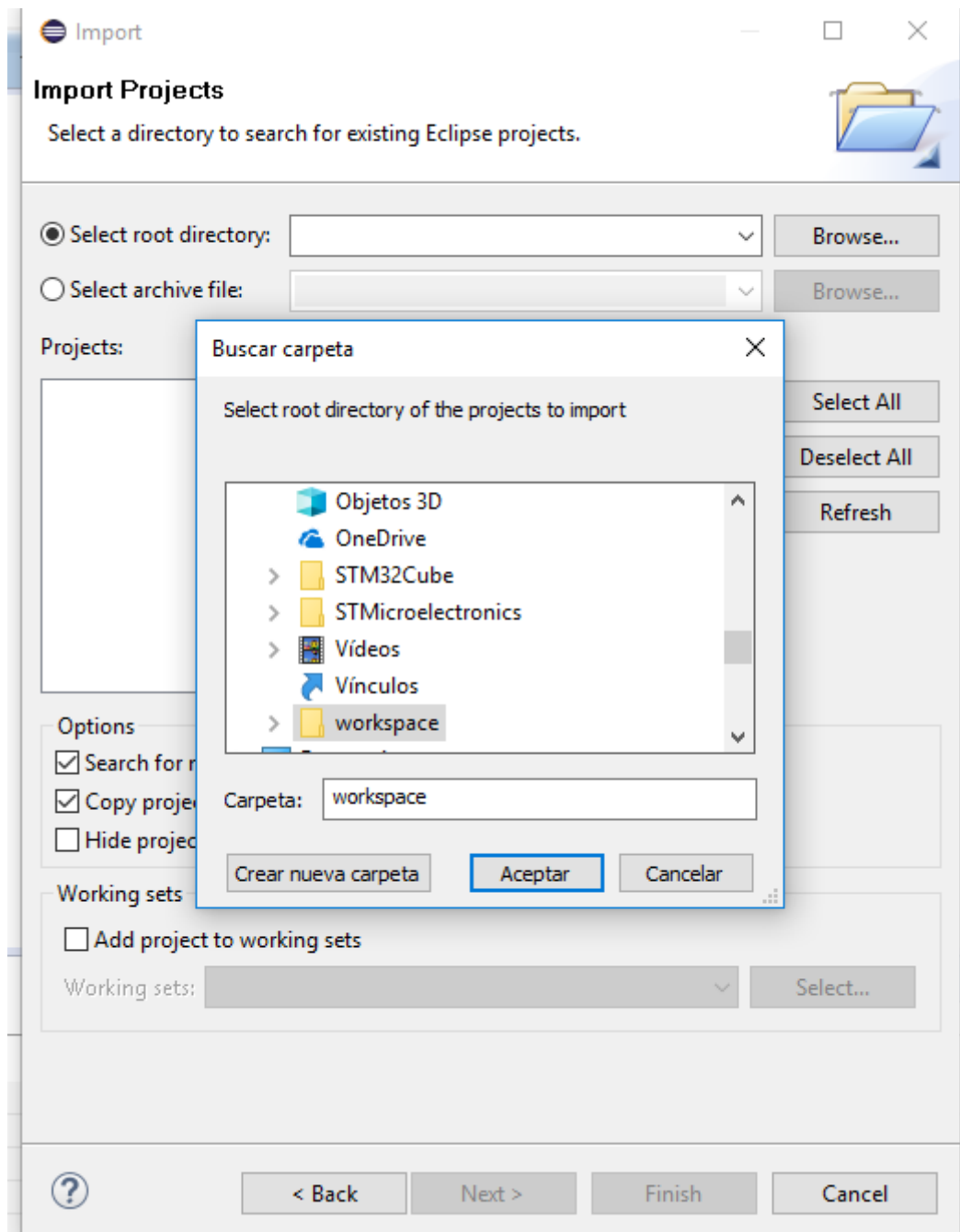
Clic en next.

Nos aparece la siguiente ventana:

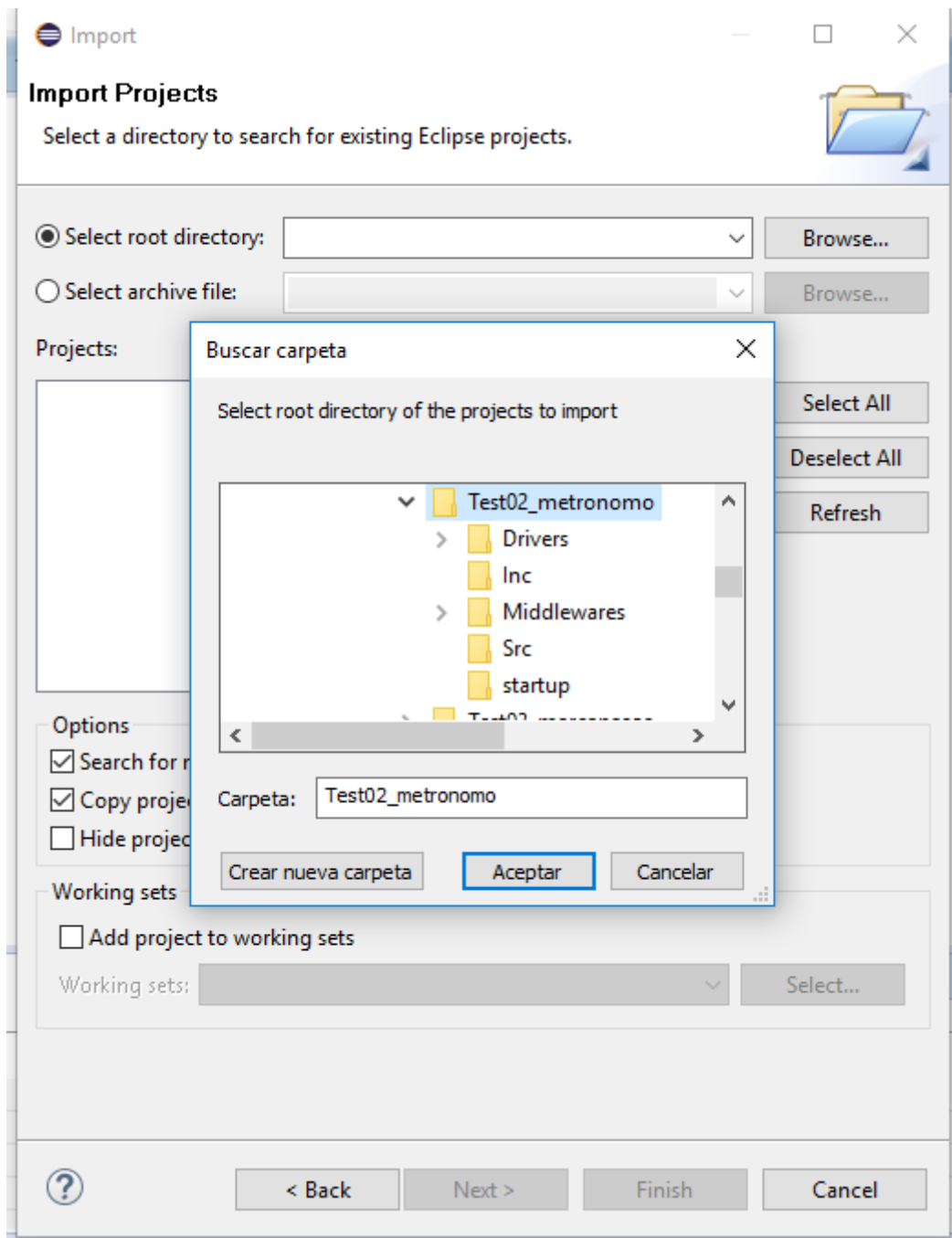


En dicha ventana clic en el botón 'Browse' señalado en la anterior figura.

Nos aparece una nueva ventana emergente:

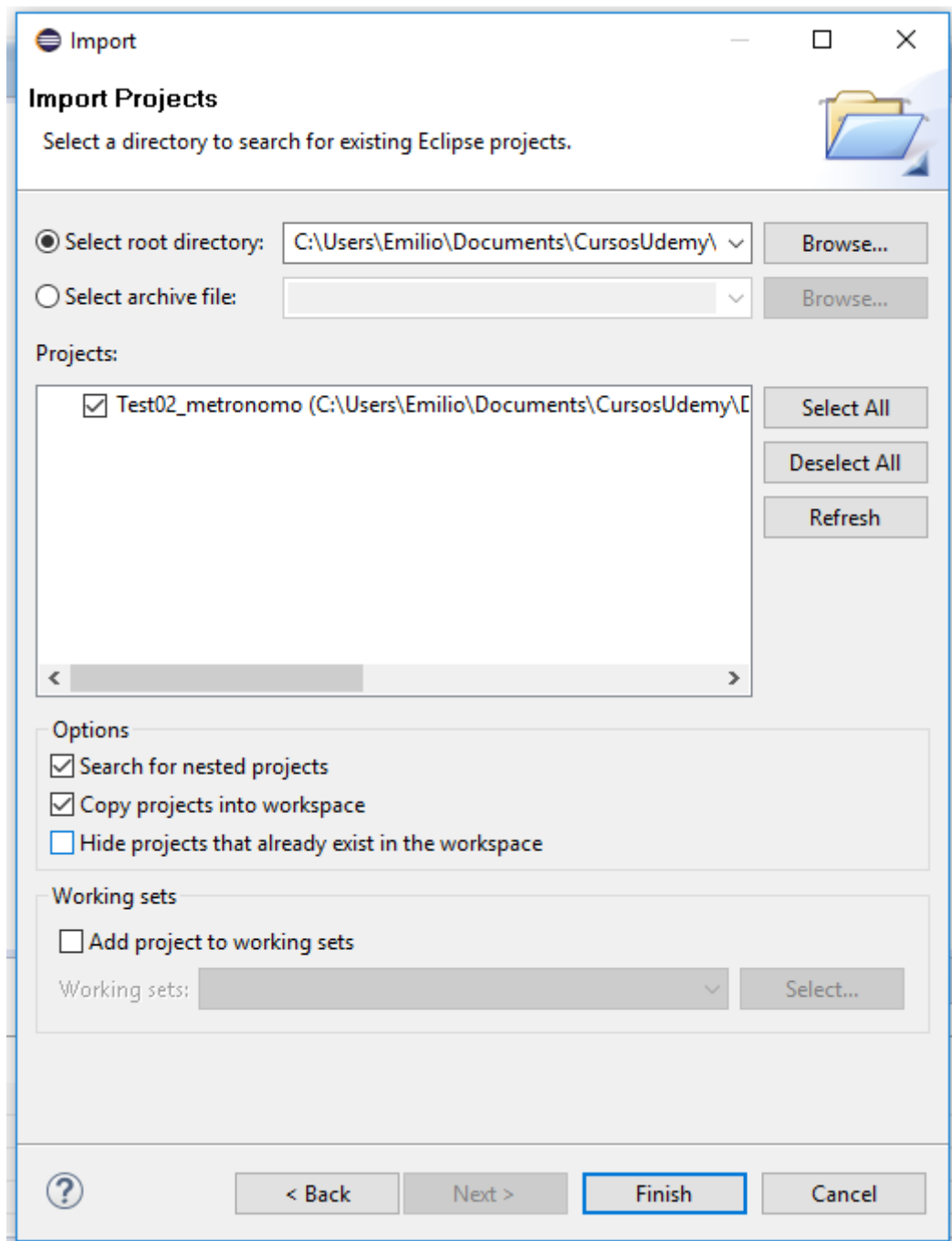


En dicha ventana navegamos hasta la ubicación del el archivo descargado y descomprimido desde mi GitHub.



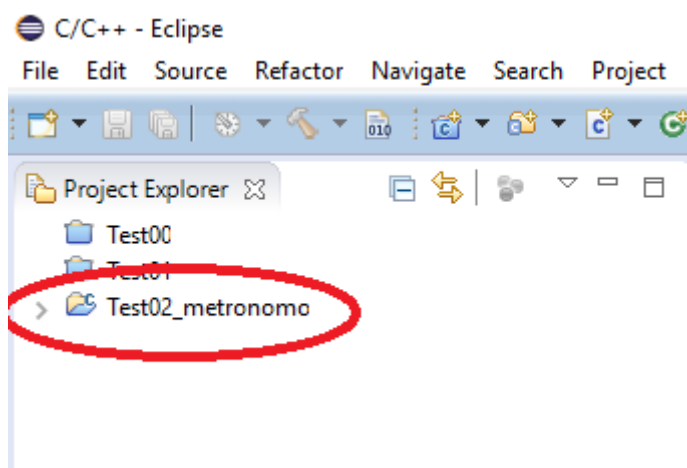
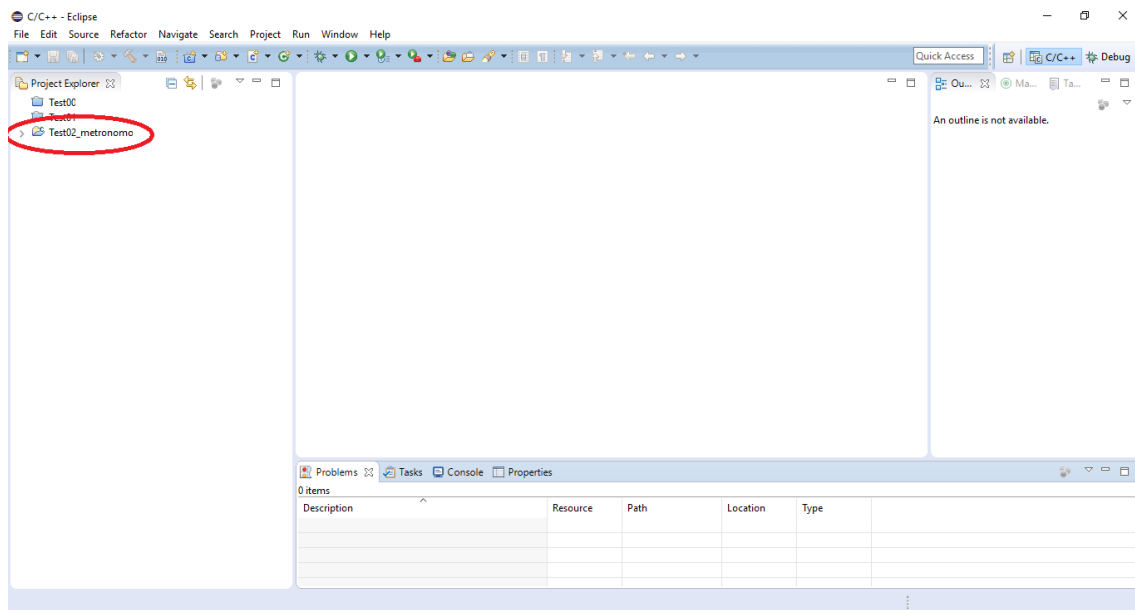
Nota: en la captura de pantalla aparece Test02_metronomo, en realidad hay que buscar el proyecto Test03_marcapasos.

Clic en aceptar y clic en finish.



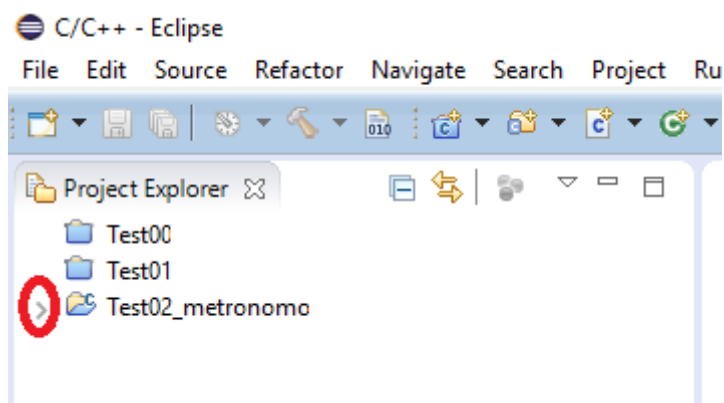
Nota: en la captura de pantalla aparece Test02_metronomo, en realidad debe aparecer el proyecto Test03_marcapasos.

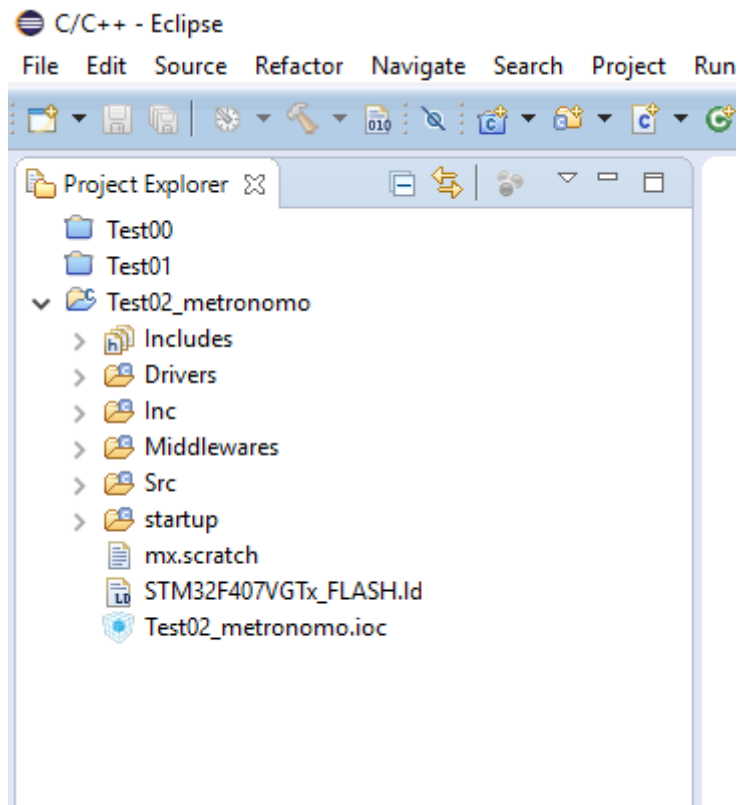
Ahora observamos que tenemos el proyecto en el panel de la izquierda "Project Explorer".



Nota: en las capturas de pantalla aparece Test02_metronomo, en realidad debe aparecer el proyecto Test03_marcapasos.

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





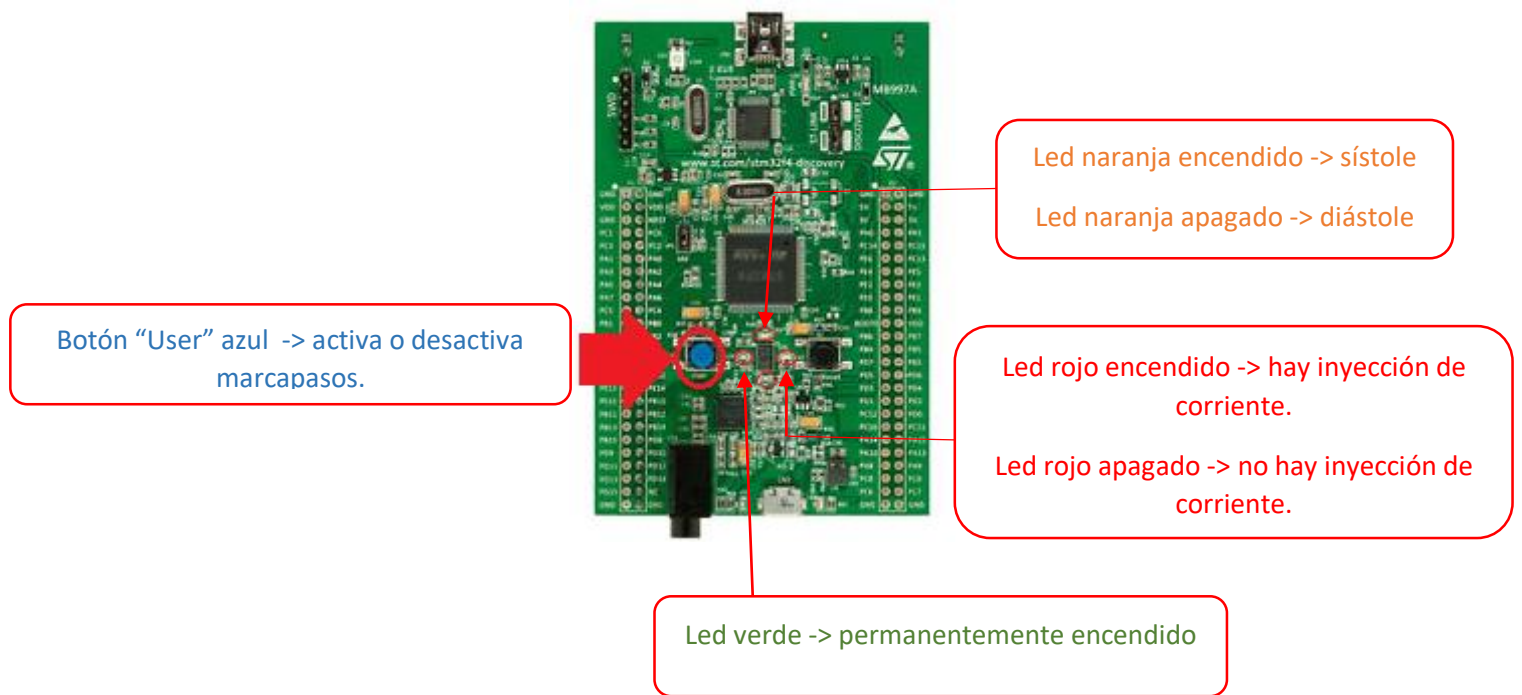
- **¿Qué hace éste programa cuando se carga en la placa?.**

Éste programa simula mediante los 2 Leds de la placa (el naranja y el rojo) el funcionamiento de un marcapasos, de forma que:

- **LED Naranja** -> representa la frecuencia de pulso cardiaco deseada. Suponemos una duración de medio segundo de sístole (corazón contraído), y medio segundo de diástole (corazón expandido).

La **sístole** se representa con el **led naranja encendido**, y la **diástole** con el **led naranja apagado**.

- **LED Rojo** -> representa el impulso eléctrico inyectado en el corazón para producir las sístoles (contracciones del corazón). Suponemos que es un impulso de corriente eléctrica de duración 10 ms.



La corriente eléctrica inyectada por el marcapasos, produce casi inmediatamente la sístole (o contracción del corazón y bombeo de sangre).

Es por ello que ambas señales se activan a la vez, y por tanto led naranja y rojo se encenderán la vez.

Sin embargo, la corriente eléctrica necesaria inyectada es sólo un impulso de pequeña duración. Estimamos que basta con un impulso eléctrico de duración 10 ms para la inyección de corriente, de forma que se mantenga una diástole de medio segundo, y luego una diástole (expansión del corazón) de otro medio segundo.

Es por ello que el Led rojo permanece encendido bastante menos tiempo que el naranja, sólo un instante.

La secuencia sería la siguiente:

Led rojo y naranja se encienden -> se inyecta corriente eléctrica y comienza la sístole (corazón se contrae).

Transcurre 10 ms.

Led rojo se apaga. -> deja de inyectarse corriente eléctrica.

Transcurre 0,5 sg

Led naranja se apaga -> se termina la sístole y empieza la diástole (el corazón se relaja y por tanto se expande).

Transcurre 0,5 sg.

Led rojo y naranja se encienden -> se inyecta corriente eléctrica y comienza la sístole (corazón se contrae).

Transcurre 10 ms.

Led rojo se apaga -> se termina la sístole y empieza la diástole (el corazón se relaja y por tanto se expande).

Transcurre 0,5 sg

y así sucesivamente.

.

.

.

.

Para iniciar el proceso anterior se pulsa el botón "User" de color azul de la placa, y para interrumpirlo se vuelve a pulsar.

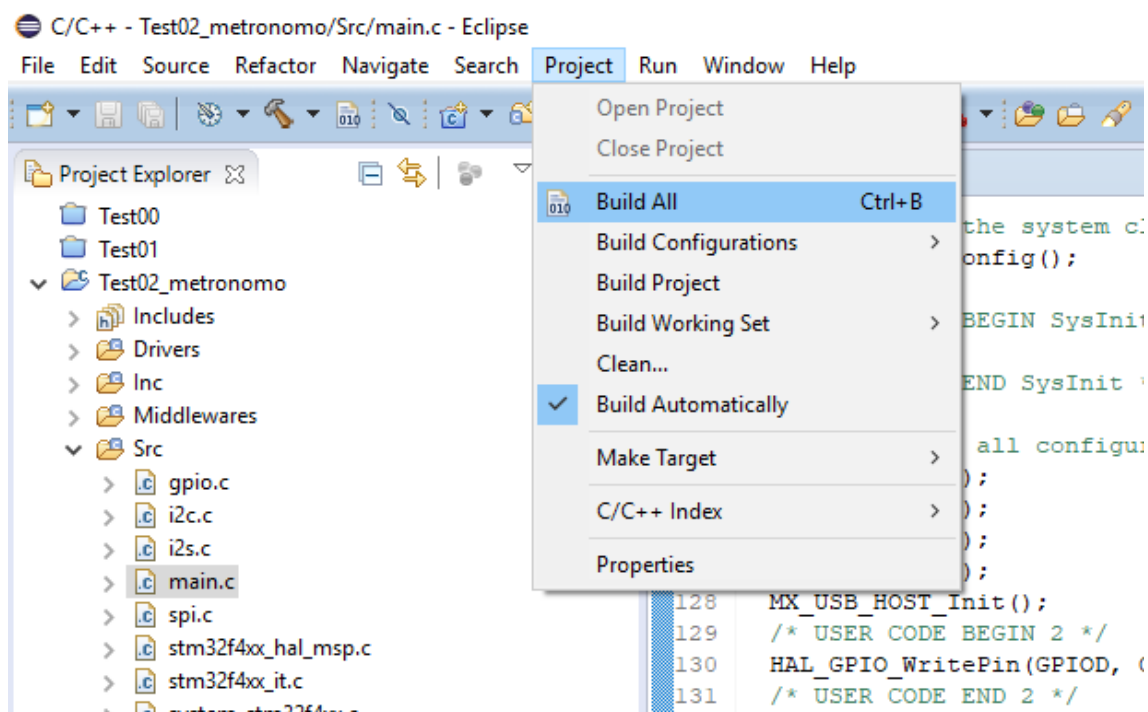
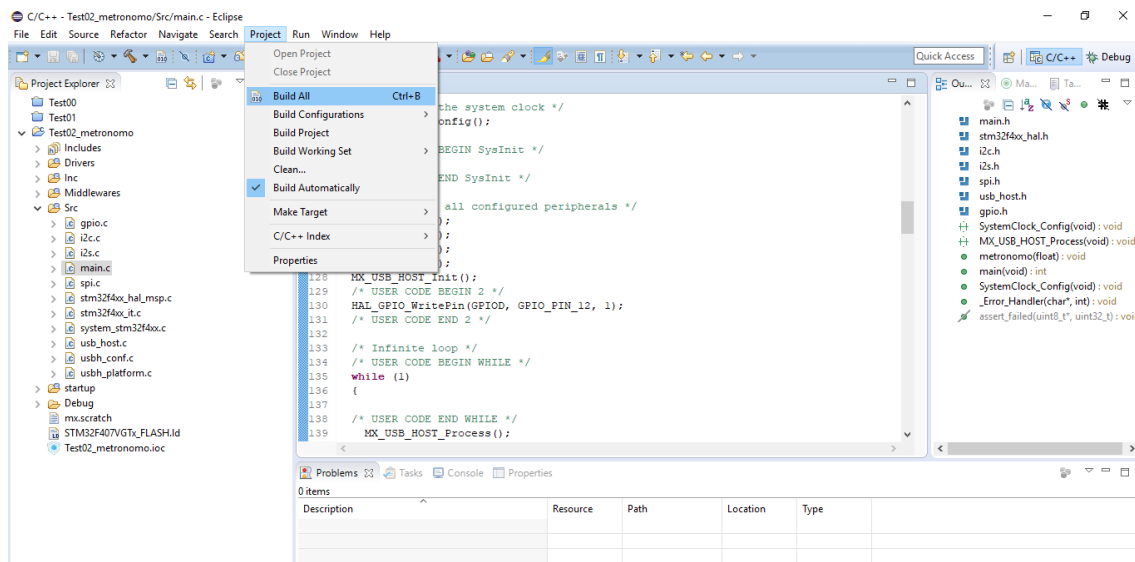
De manera que el botón "User" hace de interruptor del sistema.

El led verde estará permanentemente encendido como señal de que el programa está cargado en la placa.

• Carga y ejecución del programa en la placa.

Nota Importante: el proceso es exactamente el mismo que el usado en otro proyecto subido a éste GitHub llamado 'Test02_Metronomo', por eso he usado las capturas de pantalla de ese proyecto dónde usted verá Test02_Metronomo en lugar de 'Test03_Marcapasos' (para horrar tiempo). Siga exactamente los pasos que se indica, pero con 'Test03_Marcapasos'.

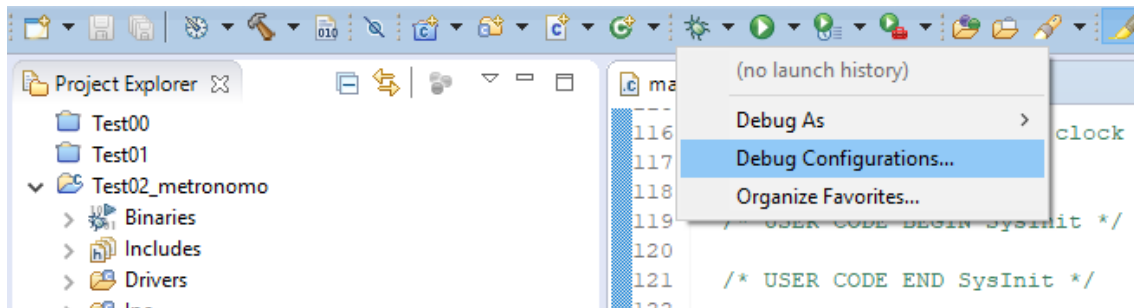
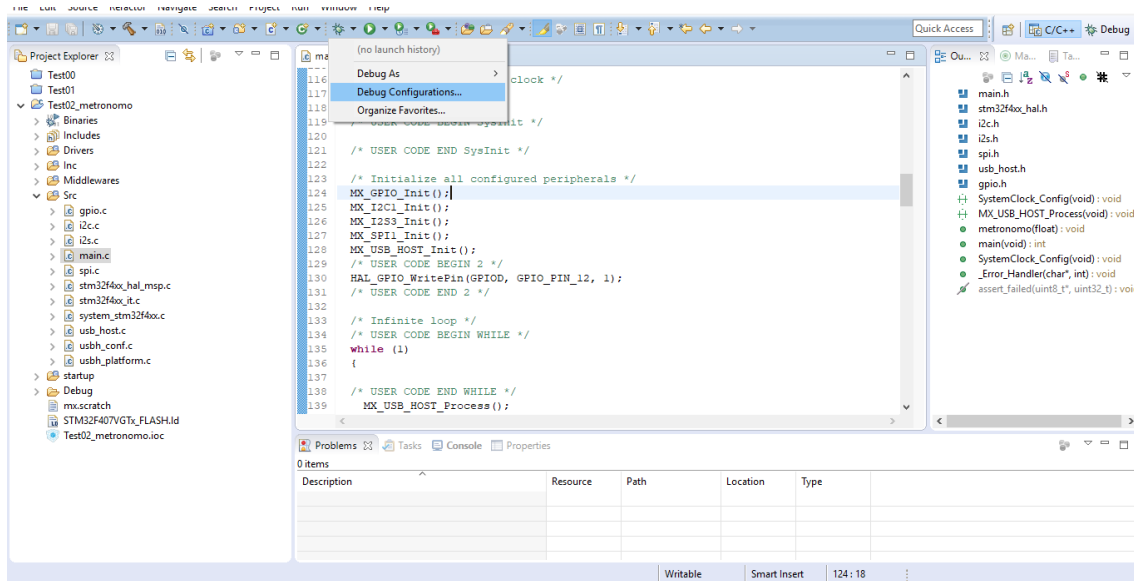
Una vez que tenemos el programa listo, lo compilamos. Para ello hacer clic en 'Project' y en el desplegable hacer clic en 'Build all'.



Nota: en la captura de pantalla aparece Test02_metronomo, en realidad debe aparecer el proyecto Test03_marcapasos.

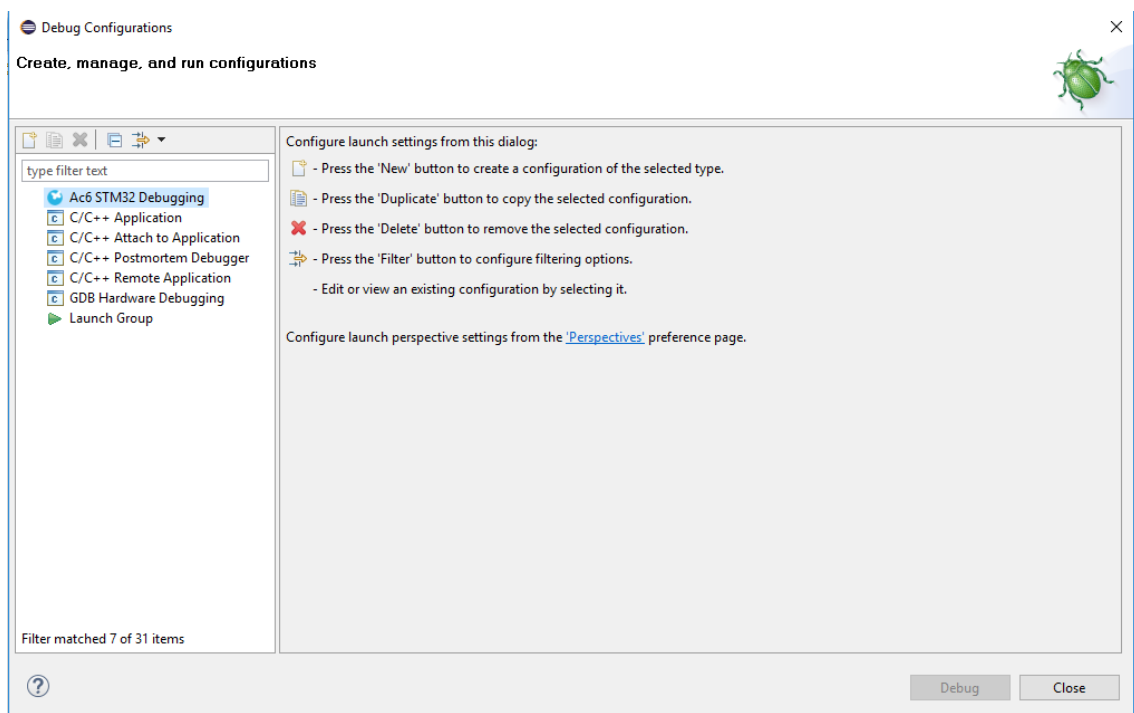
Esto generará el archivo que usará el micro para ejecutar el programa.

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

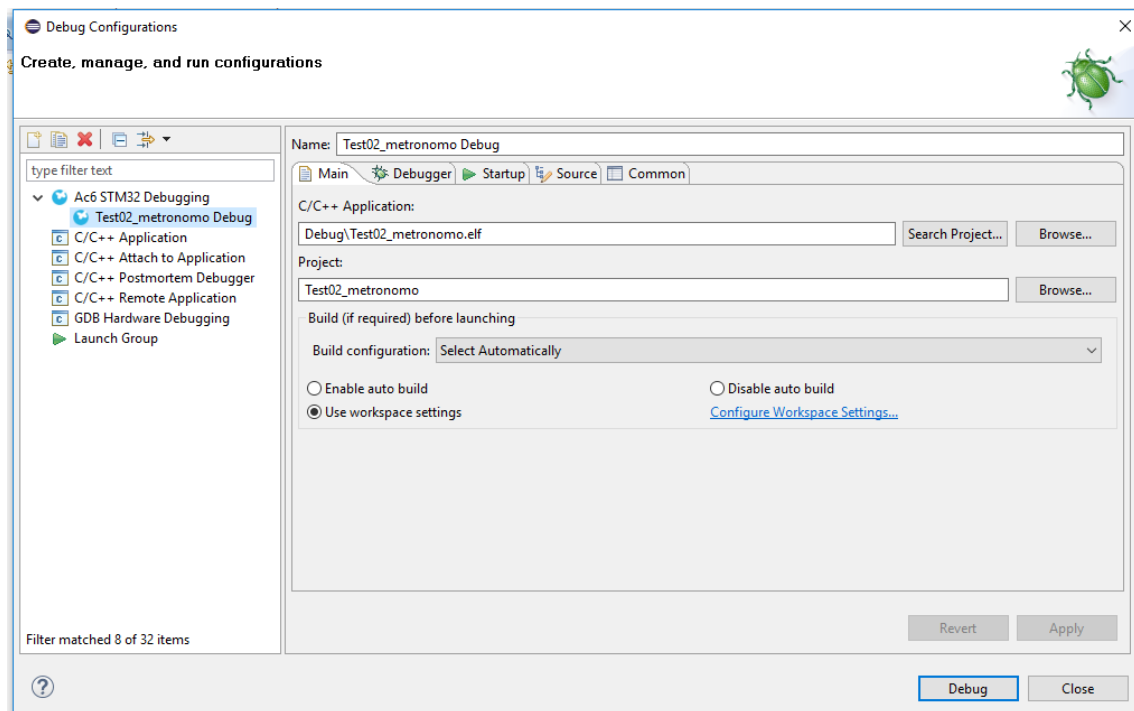


Nota: en la captura de pantalla aparece Test02_metronomo, en realidad debe aparecer el proyecto Test03_marcapasos.

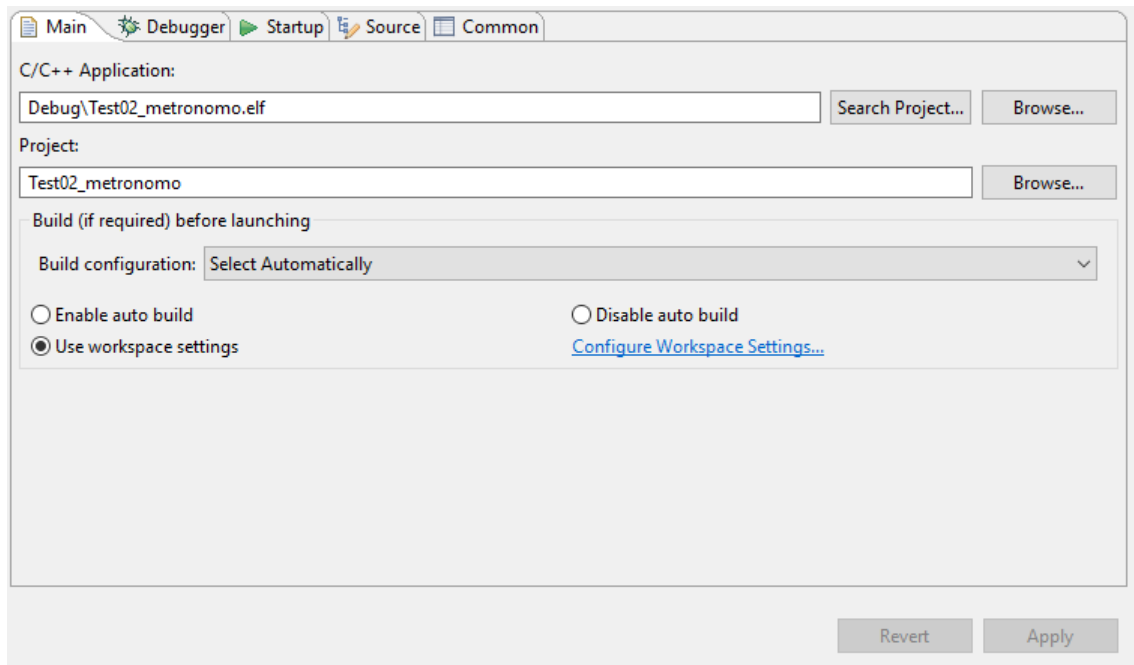
Aparece la ventana:



Hacer doble clic en 'Ac6 STM32 Debugging'.



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



Debugger Setup

GDB Command: Browse... Variables...

Command Set: Standard (Windows) v

Protocol Version: mi v

☐ Verbose console mode

OpenOCD Setup

OpenOCD Command: Browse... Variables...

OpenOCD Options:

Port number:

Script:

Revert Apply

Debug Configurations

Create, manage, and run configurations

Name: Test02_metronomo Debug

Initialization Commands

☒ Reset and Delay (seconds):

☒ Halt

Load Image and Symbols

☒ Load image

☒ Use project binary: Test02_metronomo.elf

☐ Use file: Workspace... File System...

Image offset (hex):

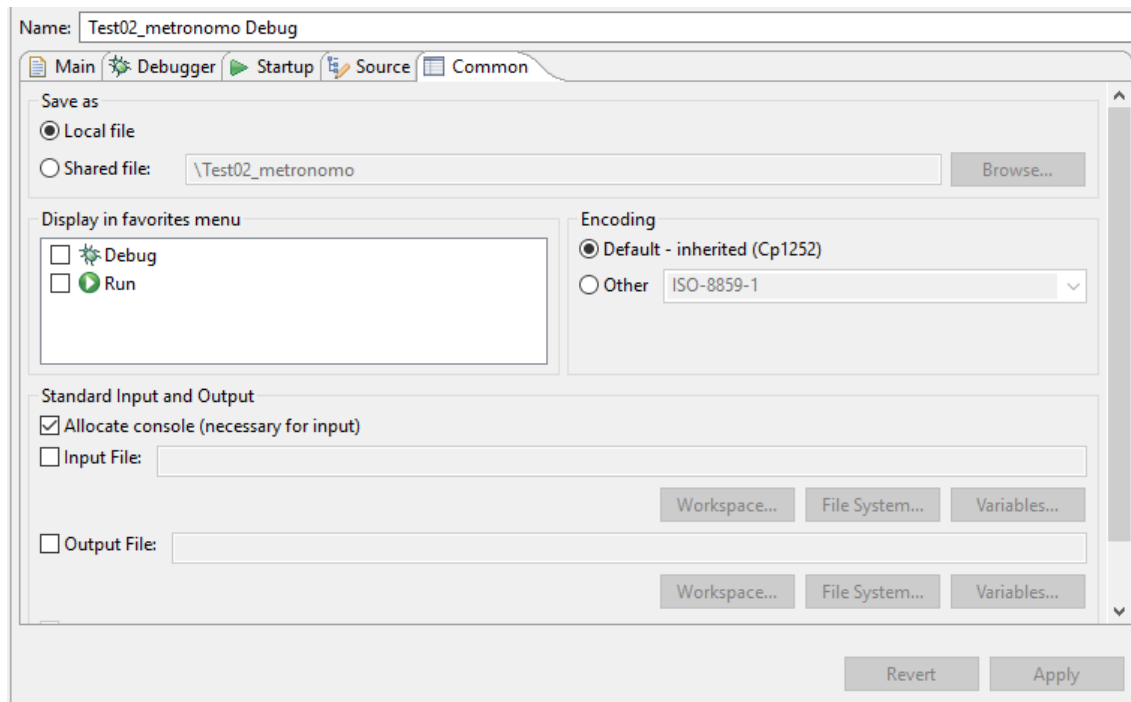
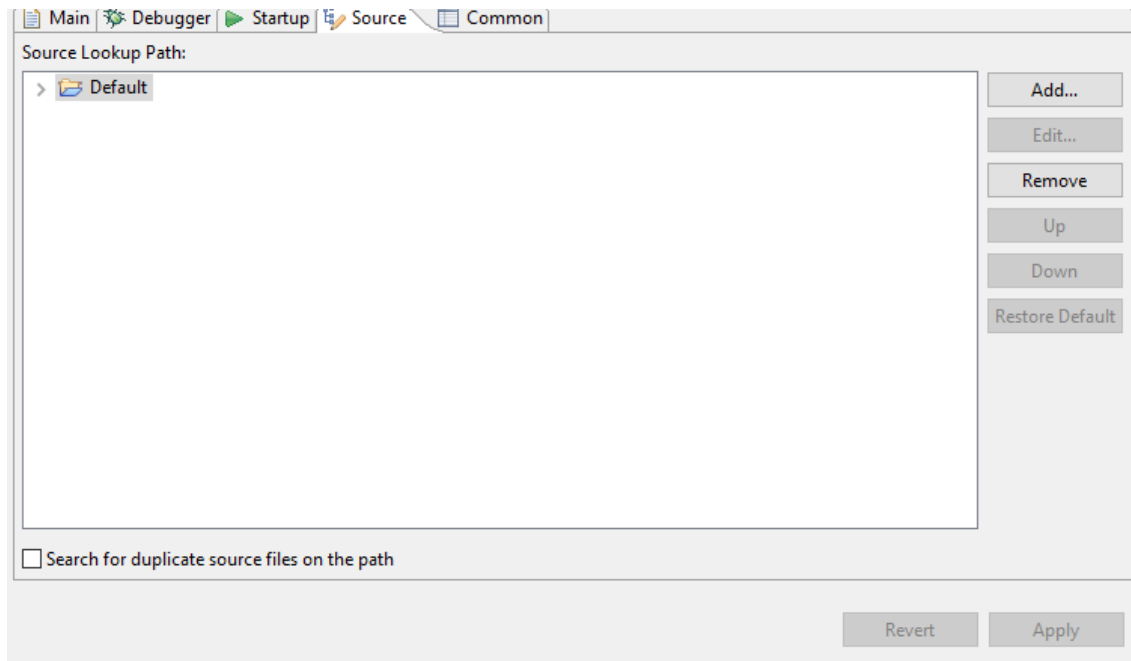
☒ Load symbols

☒ Use project binary: Test02_metronomo.elf

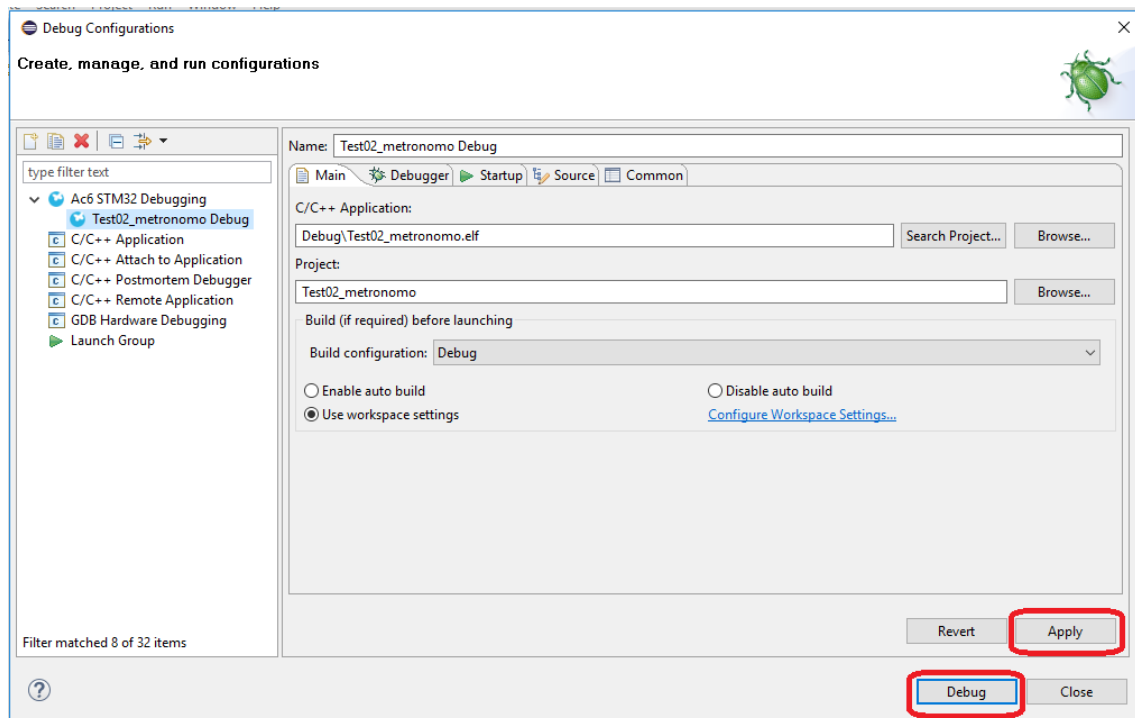
☐ Use file: Workspace... File System...

Filter matched 8 of 32 items

Debug Close

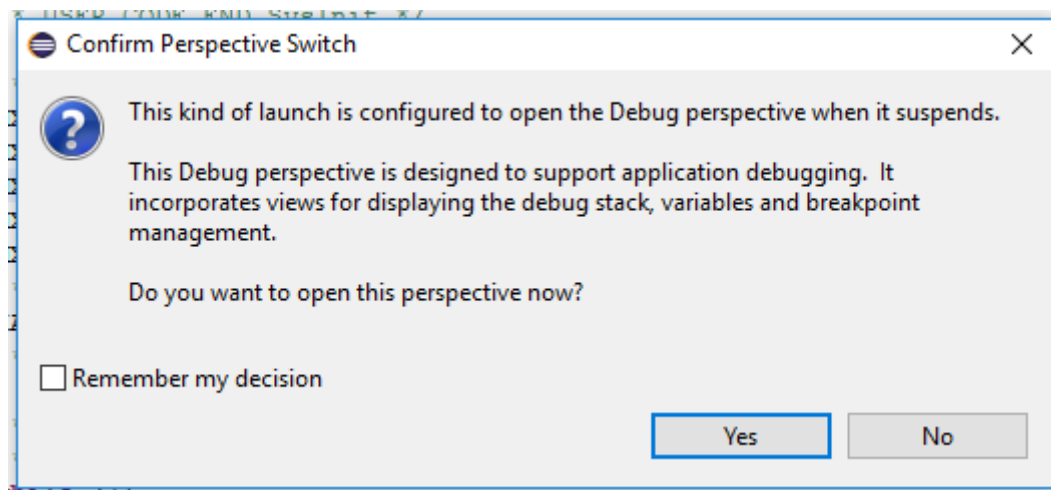


A continuación, clic en 'Apply' y clic en 'Debug'.

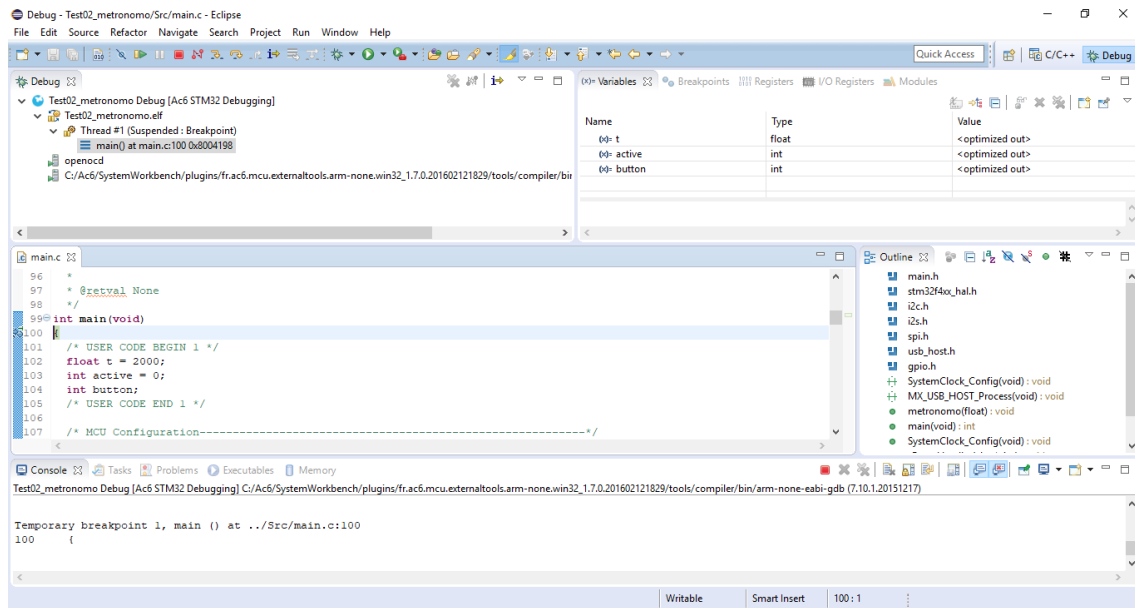


Nota: en las capturas de pantalla aparece Test02_metronomo, en realidad debe aparecer el proyecto Test03_marcapasos.

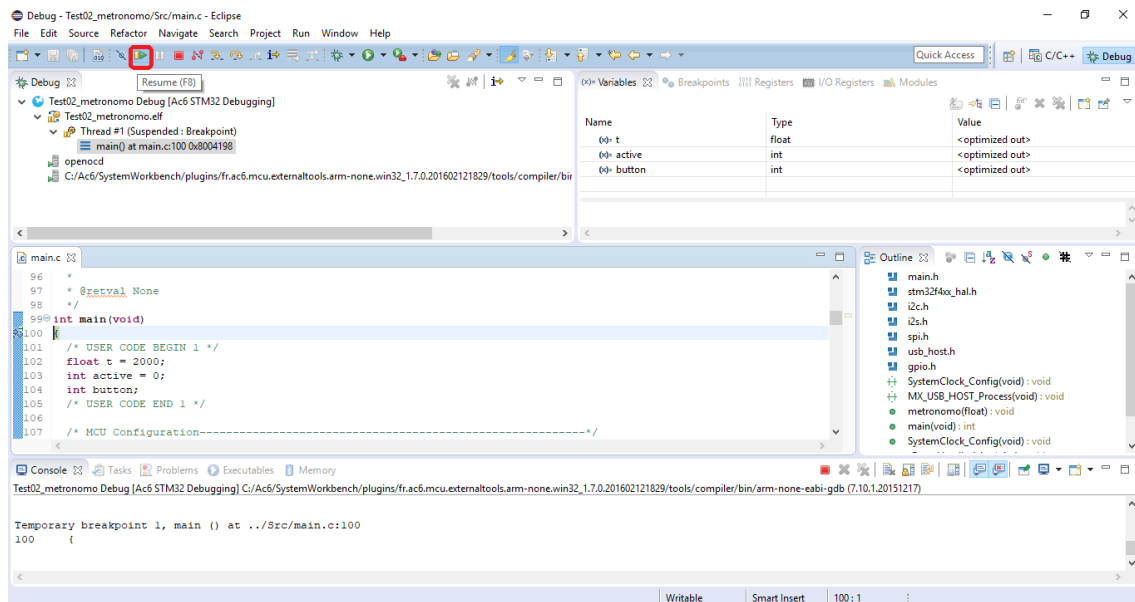
Nos aparece la siguiente ventana emergente:

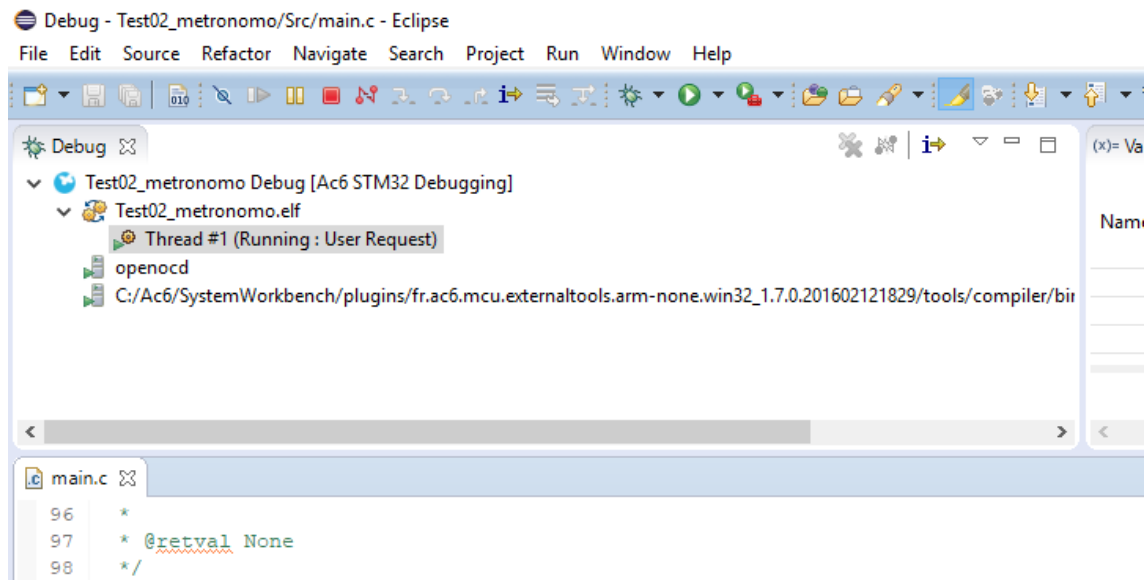
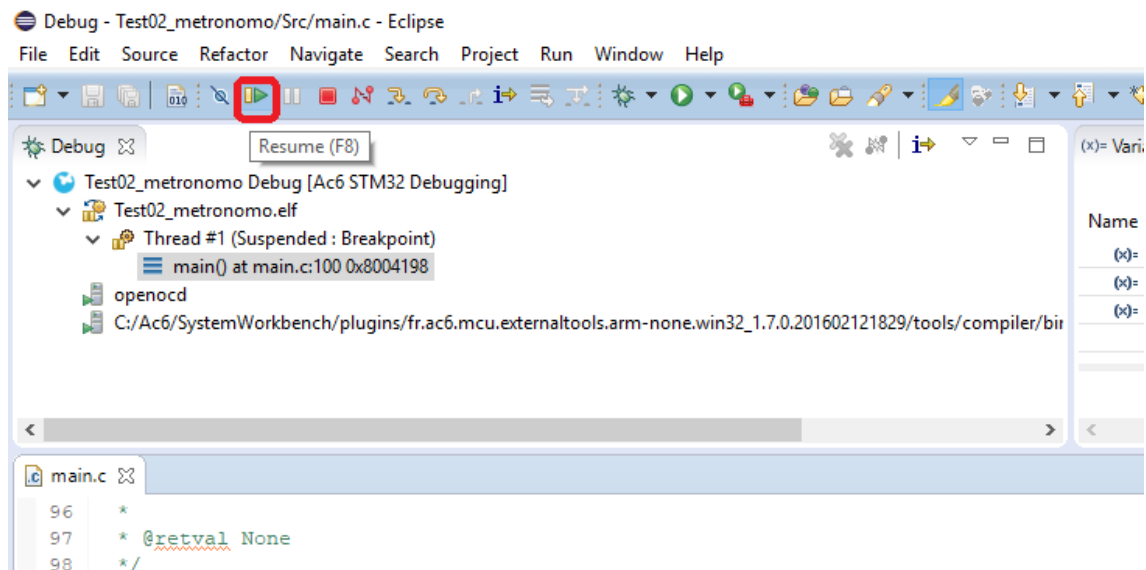


Si hacemos clic en 'Yes', tendremos la vista del programa en modo depuración. Hacemos clic en 'Yes' y tenemos:



Ahora hacemos clic en 'Resume':

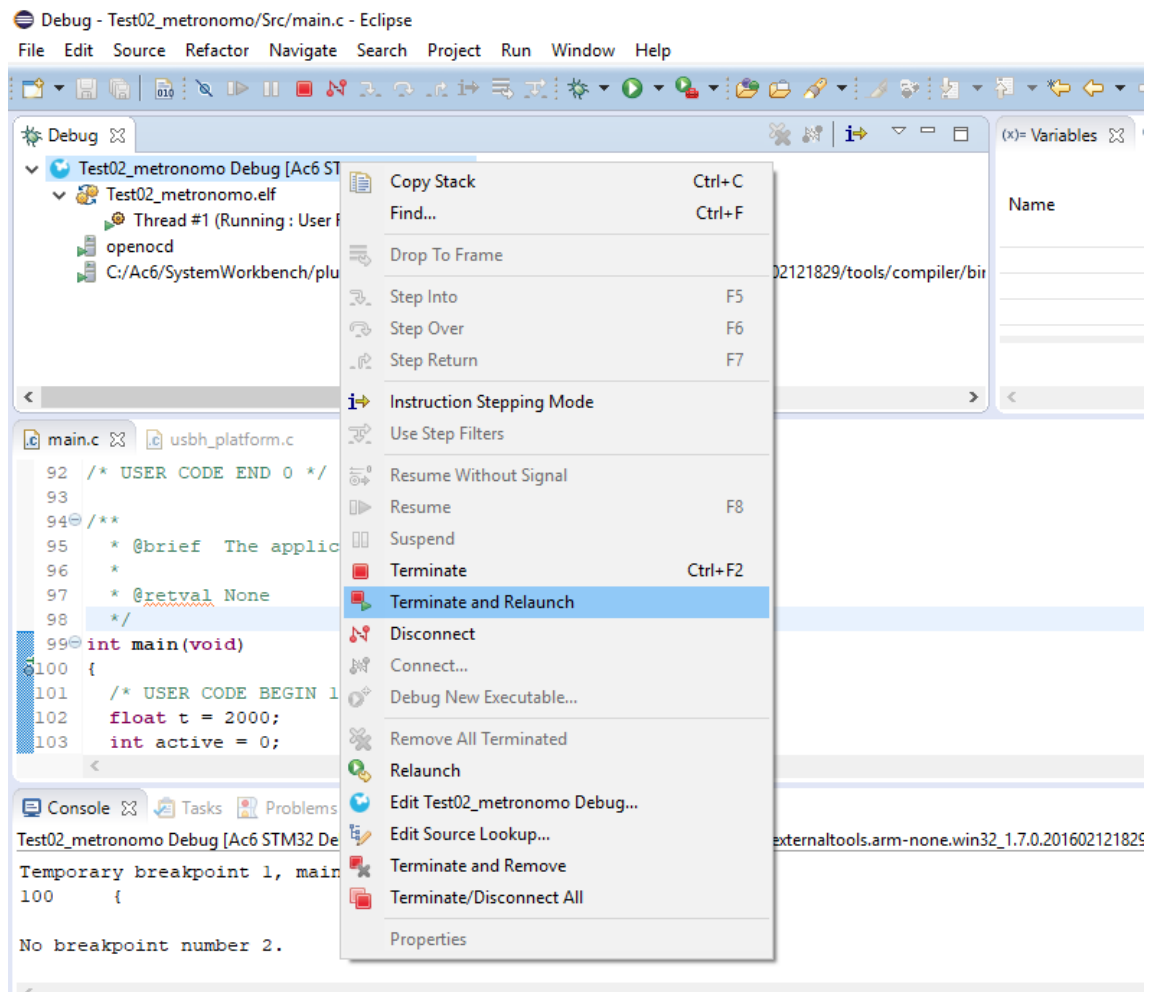




Nota: en las capturas de pantalla aparece Test02_metronomo, en realidad debe aparecer el proyecto Test03_marcapasos.

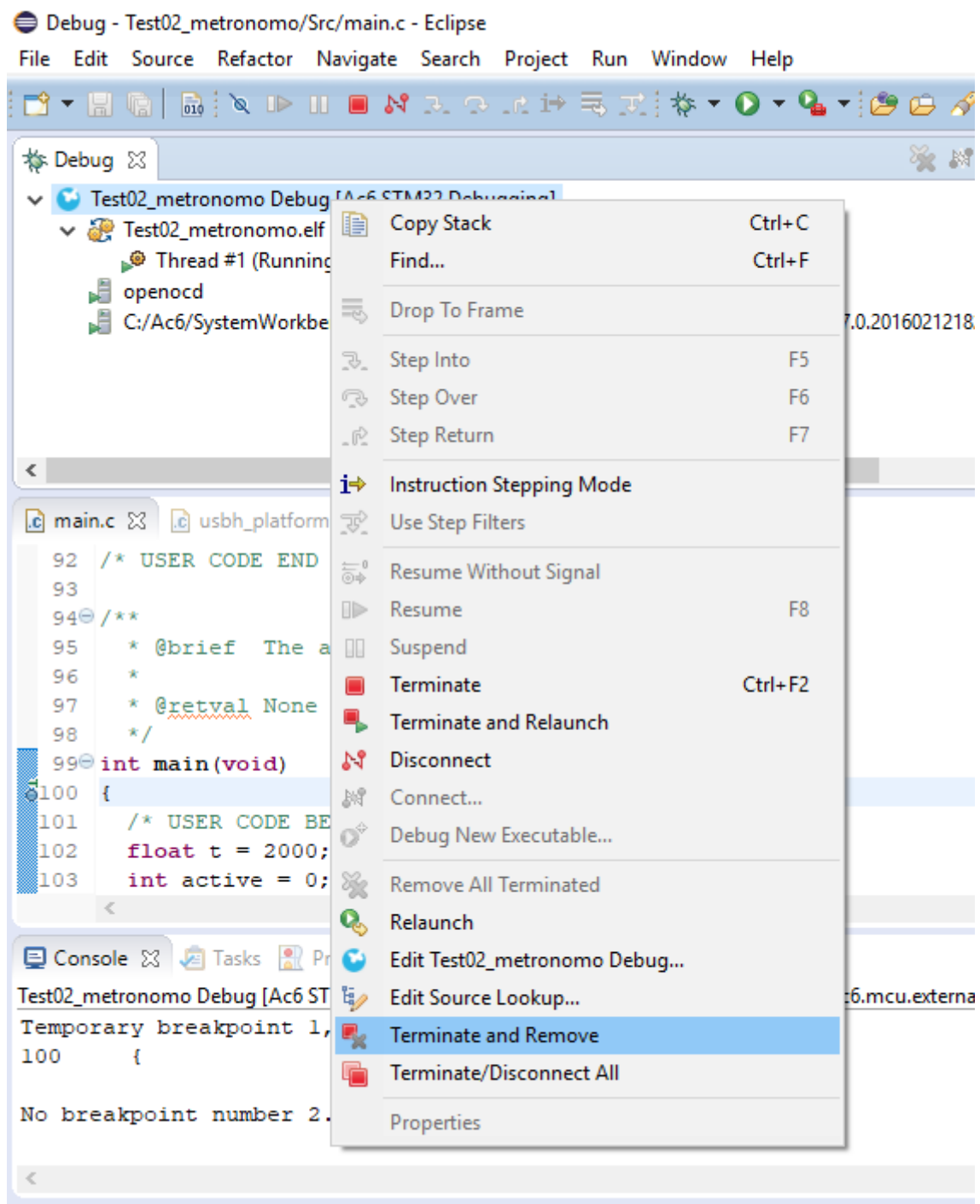
El programa ya se ha descargado en la placa y podemos pulsar el botón 'User' de color azul para comprobar su funcionamiento.

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.



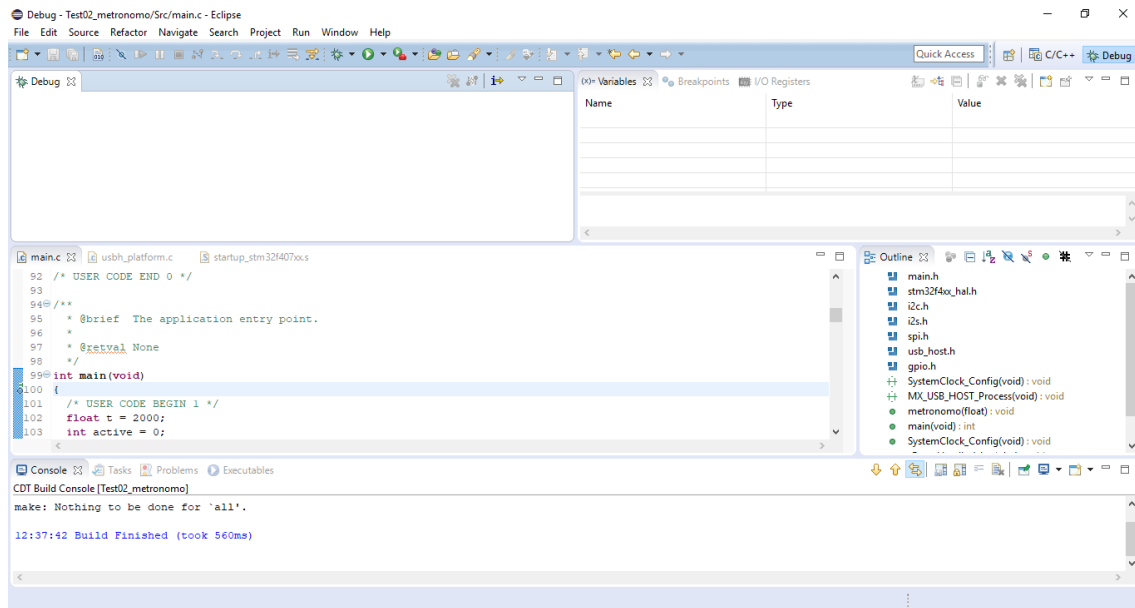
Luego podemos volver a hacer clic en 'Resume'.

Si se desea terminar y eliminar la depuración en curso:

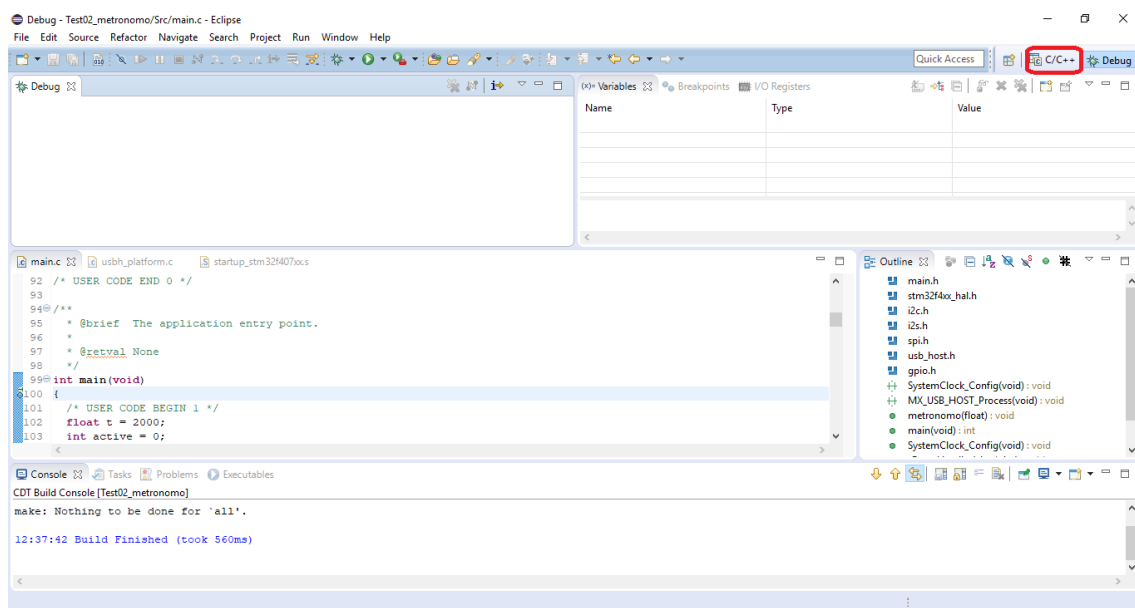


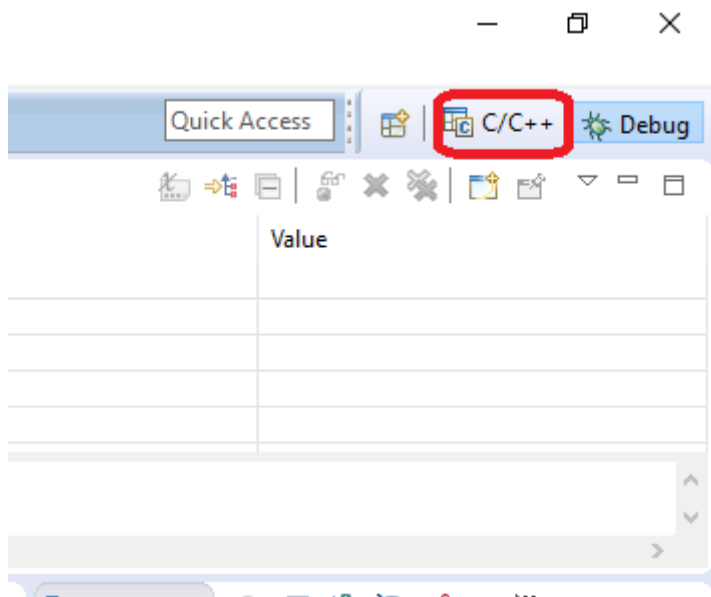
Nota: en la captura de pantalla aparece Test02_metronomo, en realidad debe aparecer el proyecto Test03_marcapasos.

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.

