TUTORIAL: PROGRAMANDO CHRONOS

Importante:

El presente tutorial sirve para instalar y configurar todas las herramientas (de software libre) necesarias para la programación y el desarrollo con la tarjeta base CHRONOS a través del puerto paralelo (método conocido como *JTAG Wiggler)* de cualquier PC con Windows XP.

Todo el software requerido para llevar a cabo este tutorial lo suministramos en una carpeta adjunta a la presente guía.

Convenciones:

(): Entre paréntesis se encuentra la versión del programa que se va a instalar y/o configurar, y que es la más reciente a la fecha en que se realiza este tutorial 11/Abril/2010

[]: Entre corchetes se encuentra una breve descripción del software que se va a instalar y/o configurar. Para mayor información referirse al marco teórico del documento principal del trabajo de grado.

↵**:** Se refiere a pulsar la tecla ENTER

Sugerencia:

Se recomienda al usuario no dejar espacios en la ruta de la ubicación de TODAS las instalaciones de los programas y archivos que aquí se presentan, de manera que la ruta que por lo general aparece por defecto en las instalaciones (C:\Archivos de programa\....) no es recomendable usarla. Para este tutorial usaremos como ejemplo la ruta C:\CHRONOS\ para instalar todos los programas que se van a presentar a continuación.

1) **JAVA** (1.6.0_19-b04) [Necesario para ejecutar aplicaciones escritas en java]**:** Verificar que este instalado y que la versión sea 1.6.0_18-b07 o posterior.

   Abrir el SIMBOLO DEL SISTEMA

   Inicio -> Accesorios -> SIMBOLO DEL SISTEMA

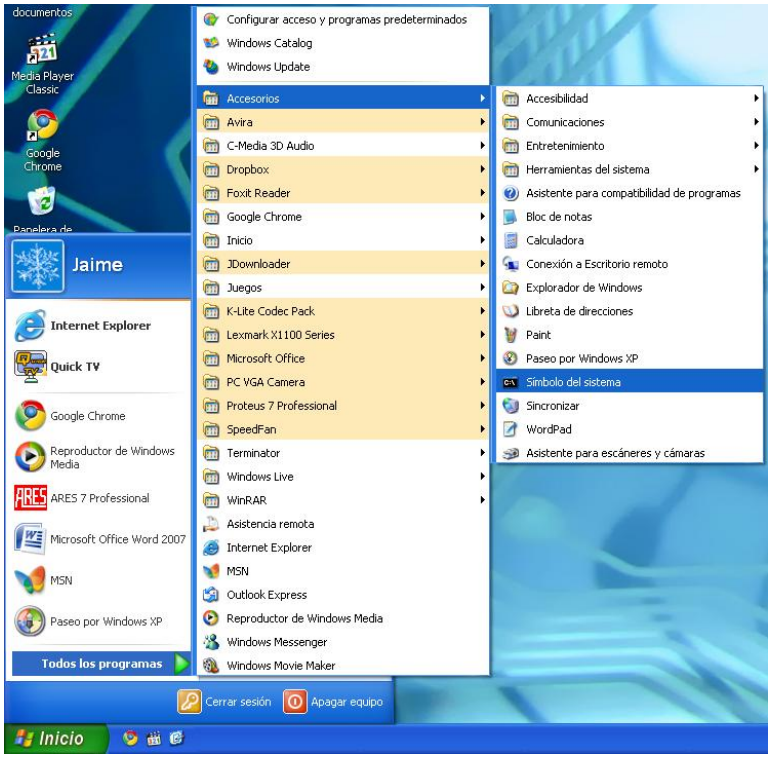


Figura 1: Accediendo al SIMBOLO DEL SISTEMA

En el escribir: *>java –version* ↵

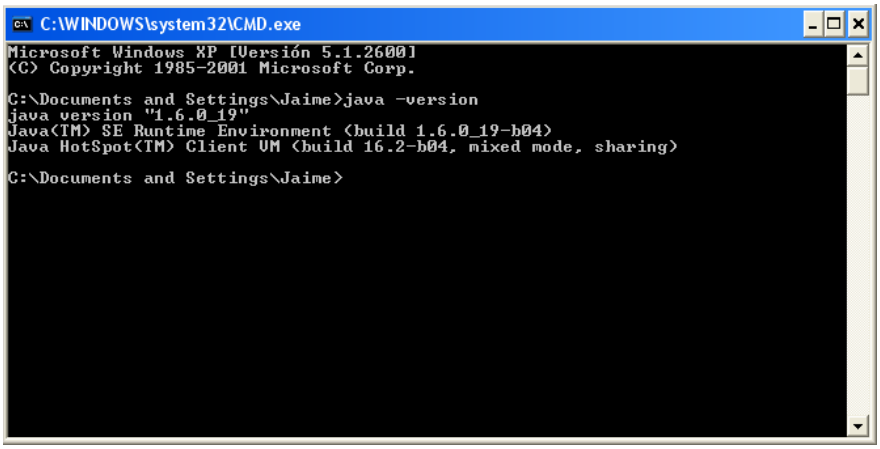


Figura 2: Confirmando la versión de JAVA del PC

En caso de no tenerlo instalado ejecutar el correspondiente instalador, y luego dar click en instalar.

2) OPENOCD (2007 re141) [Configura protocolo de comunicación JTAG]**:** Instalar.

Ejecutar el instalador correspondiente y seguir los pasos de la instalación tal y como se muestra en la figura 3.

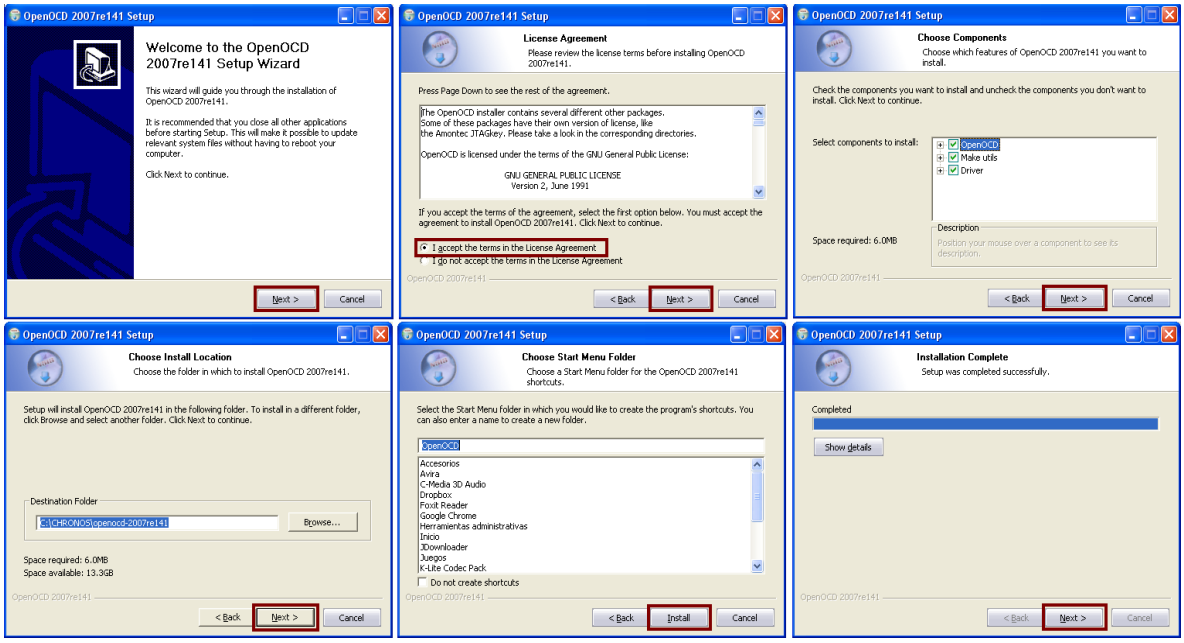


Figura 3: Pasos para la instalación del OPENOCD

Luego seleccionar finish para terminar la instalación.

3) Instalación del controlador del puerto paralelo:

Abrir EL SIMBOLO DEL SISTEMA como se indica en la figura 1, y escribir:

>*cd* (copiar luego del espacio la ruta donde se instaló el OPENOCD, inmediatamente seguida por \driver\parport) ↲

>*dir* ↲ [Para verificar que exista el archivo install_giveio.bat en la carpeta correspondiente]

>*install_giveio.bat* ↲

Si el procedimiento se realizó correctamente aparece al frente de cada proceso un *ok* y la última línea es *success*.

Para nuestro caso la ventana del símbolo del sistema para cada uno de los pasos queda como se aprecia en la figura 4. Luego cerrar la ventana del símbolo del sistema para terminar el procedimiento

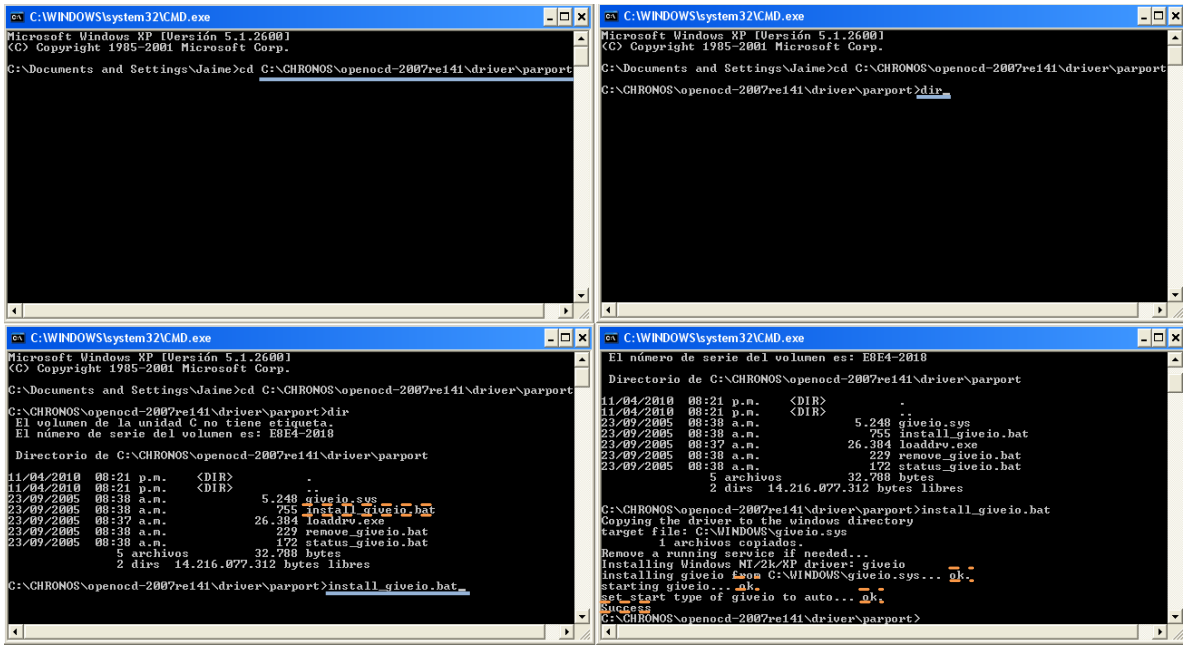


Figura 4: Procedimiento para instalar el controlador del puerto serial

4) Yagarto [Compilador *.c/.cpp -> *.hex/.bin]:

4.1) Yagarto (bu 2.20): Instalar

Ejecutar el instalador correspondiente y seguir los pasos según lo muestra la figura 5.

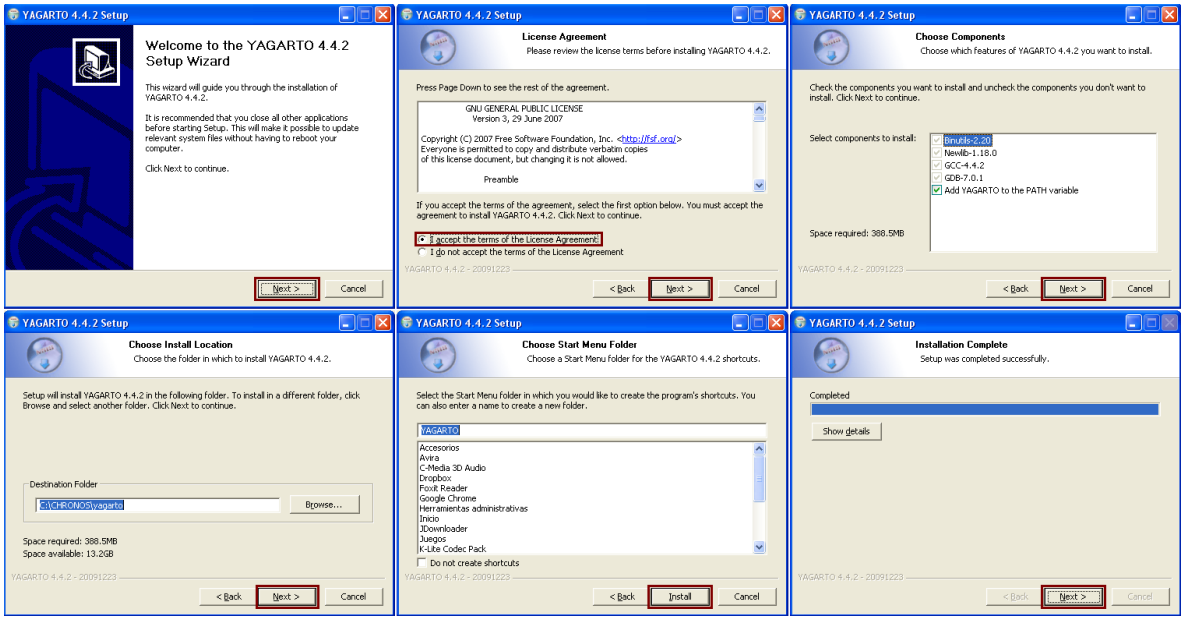


Figura 5: Pasos para la instalación del YAGARTO

Luego seleccionar finish para terminar la instalación.

4.2)YAGARTO TOOLS (20091223):Instalar

Ejecutar el instalador correspondiente y seguir los pasos según lo muestra la figura 6.

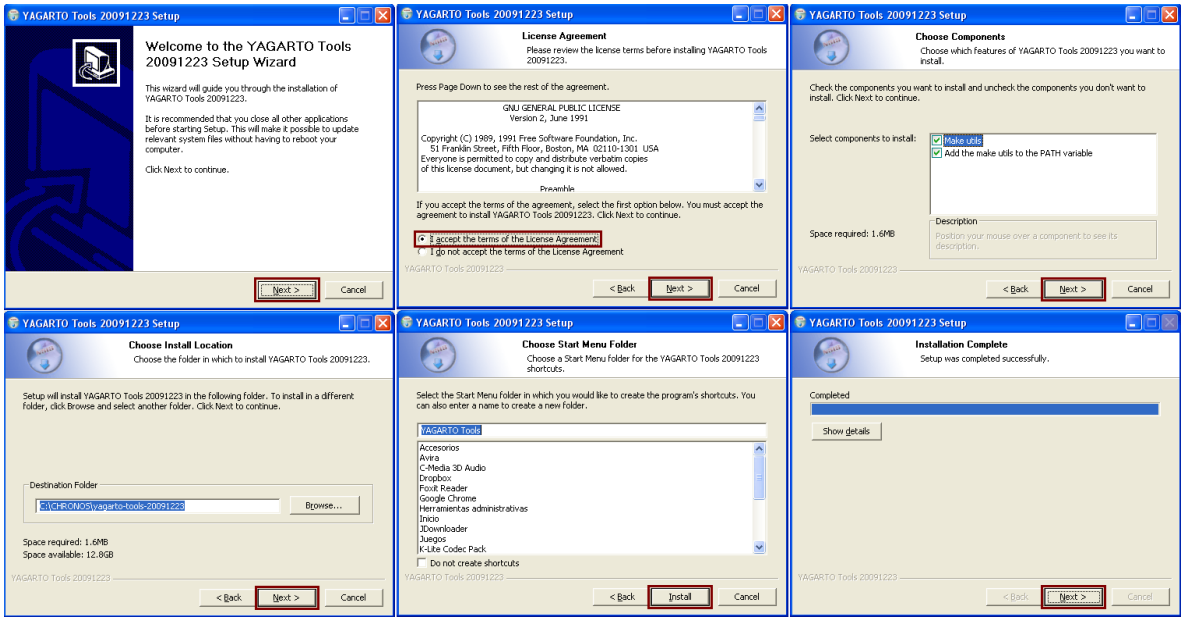


Figura 6: Pasos para la instalación del YAGARTO TOOLS

Luego seleccionar finish para terminar la instalación.

5) Eclipse (galileo 3.5.2) [es una plataforma de desarrollo opensource basada en Java]: No requiere instalación, basta con copiar la carpeta de eclipse al directorio o ubicación donde hemos instalado los programas anteriores.

5.1) Configuración Eclipse ¡Requiere conexión a internet! [Necesaria para el desarrollo de proyectos en C/C++]**:**

→ Hacer doble click sobre el archivo *eclipse.exe* presente en la carpeta eclipse.

Figura 7: Ejecutando eclipse

→ Escoger la ubicación del *workspace* teniendo en cuenta la sugerencia suministrada al comienzo del tutorial, y ojalá siendo esta la misma ubicación de los programas instalados anteriormente, tal y como se observa en la figura 8.

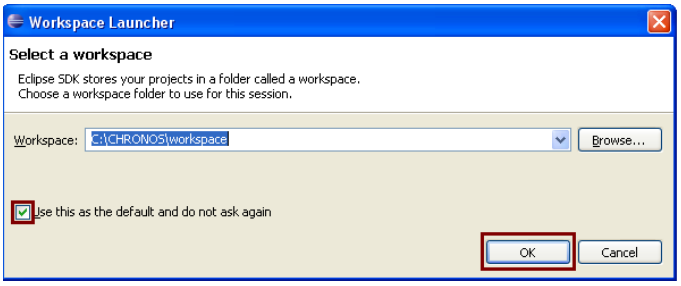


Figura 8: Eligiendo la ubicación del workspace

→ Al entrar por primera vez se muestra la pagina de bienvenida la cual cerramos seleccionando la X marcada en la figura 9

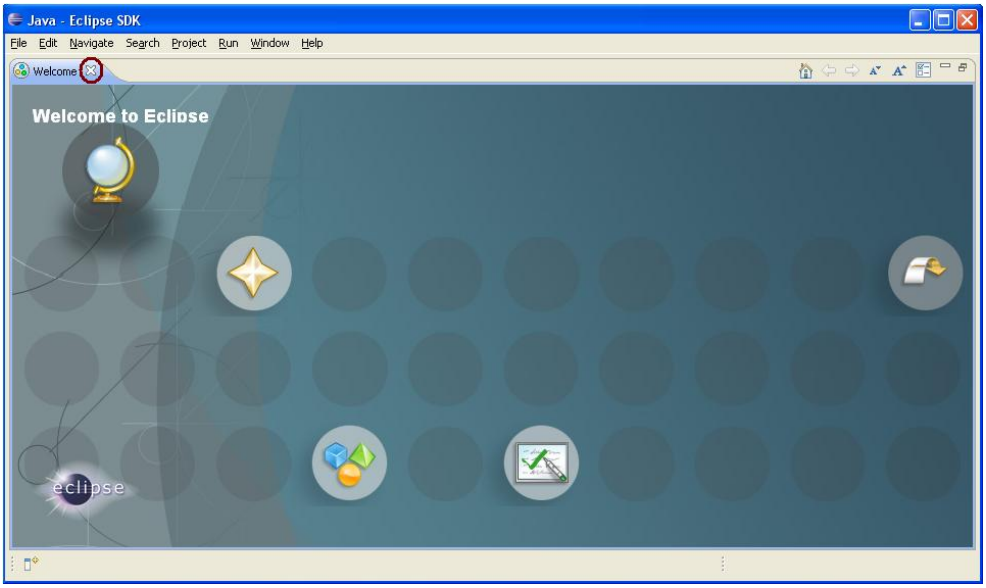


Figura 9: Página de bienvenida de Eclipse

→ Estando ya en el workbench de eclipse vamos a Help -> Install new software [Esto lo hacemos para instalar el plug-in necesario para la comunicación con dispositivos externos tipo ARM]

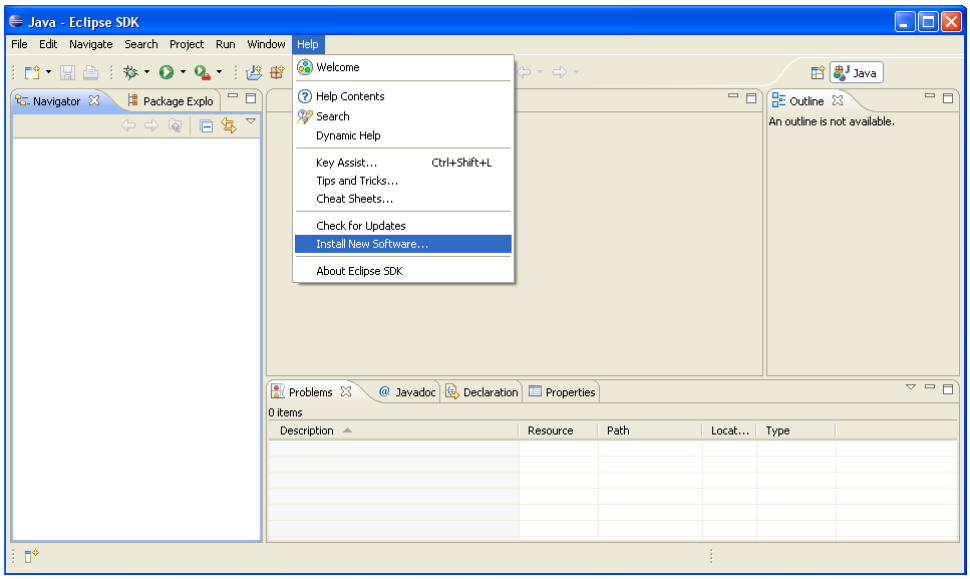


Figura 10: Instalando nuevo software en ECLIPSE

→ En la ventana que se abre (Figura 11) dar click en Add

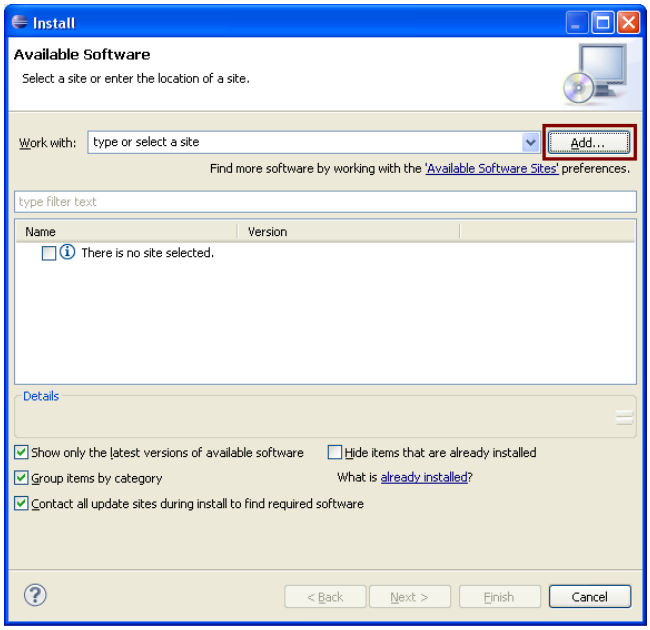


Figura 11: Ventana Install de Eclipse

→ En la nueva ventana que se abre se debe colocar *zylindcdt* en el campo name, y en el campo location colocar *http://opensource.zylin.com/zylincdt*, por último dar click en ok para cerrar la ventana.

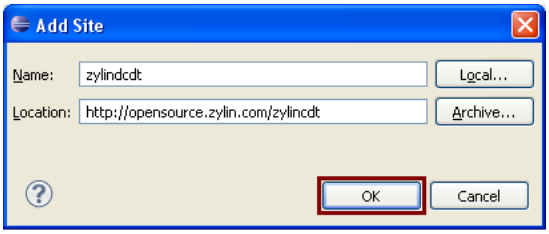


Figura 12: Ventana Add Site

Esperamos un momento hasta que aparezca en la lista el plugin *zylin embedded CDT* y lo marcamos, verificamos también que las tres opciones de la parte inferior izquierda de la ventana estén marcadas. Luego seleccionamos Next.

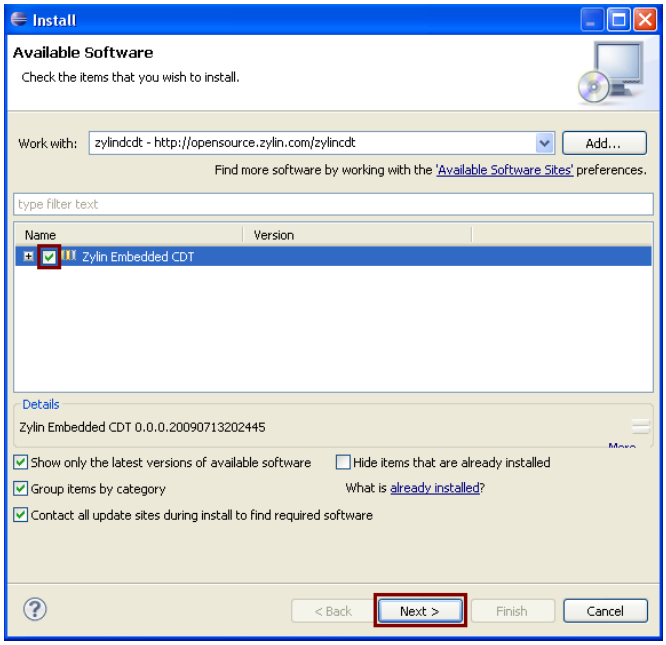


Figura 13: Ventana Install de eclipse con el Zylin embedded CDT para instalar

→ Nos muestra los detalles de la instalación, debemos dar click en next

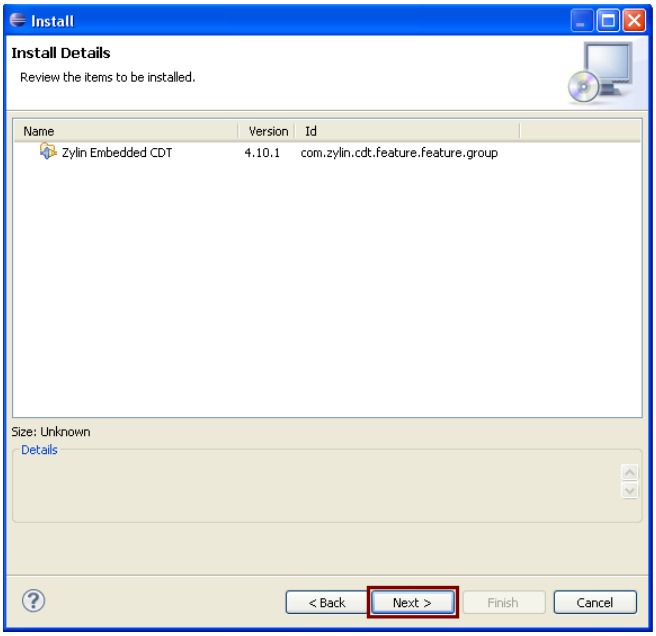


Figura 14: Ventana de los detalles de la instalación

→ Aceptamos los términos de la instalación y luego click en finish.

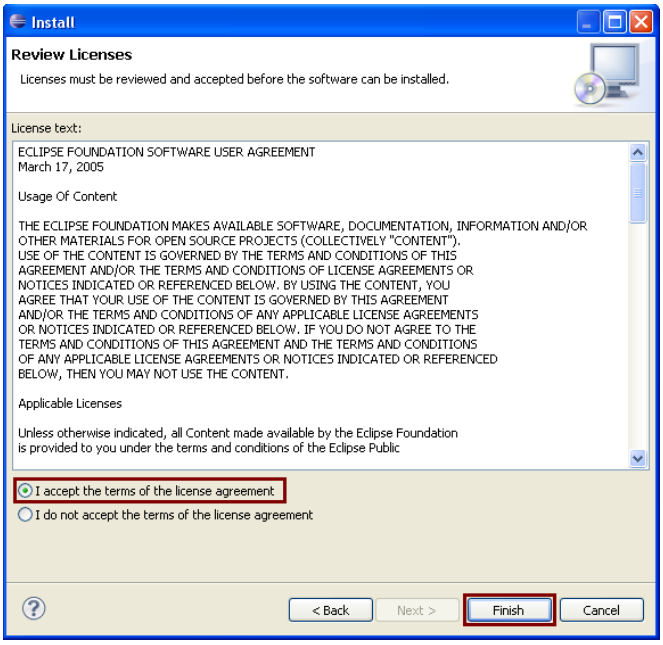


Figura 15: Aceptando los términos de la instalación

→ Durante la instalación nos puede aparecer una advertencia de seguridad sobre el contenido que vamos a instalar, este software es de confianza y por tanto pulsamos ok en la ventana de advertencia como se nota en la figura 16.

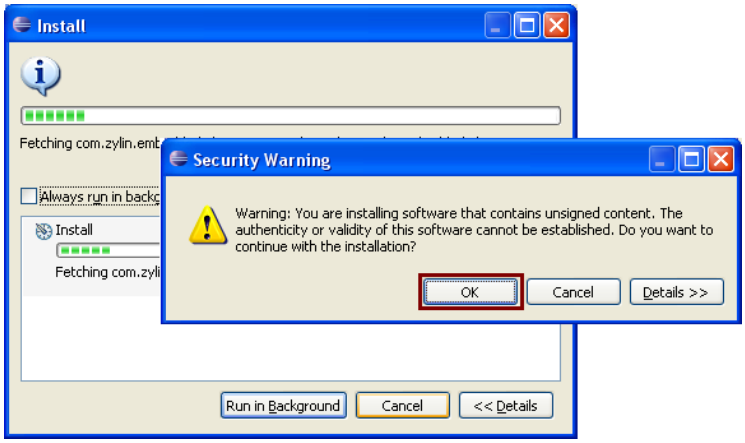


Figura 16: Advertencia de seguridad sobre el contenido que se está instalando

→ Debido a la advertencia de seguridad, nos pregunta si confiamos en el certificado de Eclipse.org Foundation, dado a que es la empresa que desarrolla el eclipse podemos confiar en ella, y por tanto seleccionamos el certificado en la lista y luego click en ok.

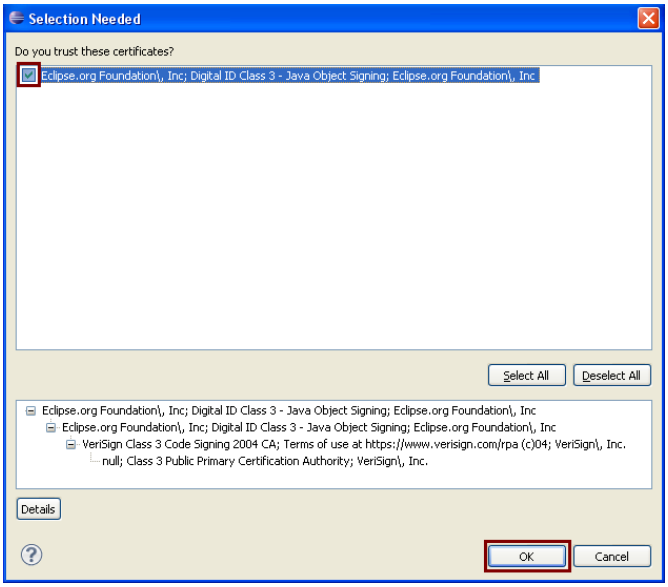


Figura 17: Ventana de selección del certificado

→ Al terminar la instalación reiniciamos el eclipse dando click en yes, para de esta forma completar correctamente la instalación.

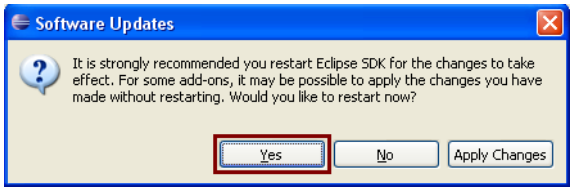


Figura 18: Ventana para reiniciar eclipse después de la instalación

5.2) Configuración de OPENOCD como herramienta externa [Para programar el microcontrolador desde eclipse]:

Antes de empezar con la configuración, es necesario copiar todos los archivos *.cfg* que se encuentran en la carpeta de instaladores, a la carpeta *bin* del directorio donde previamente se instaló el OPENOCD, tal y como se ve en la figura 19.

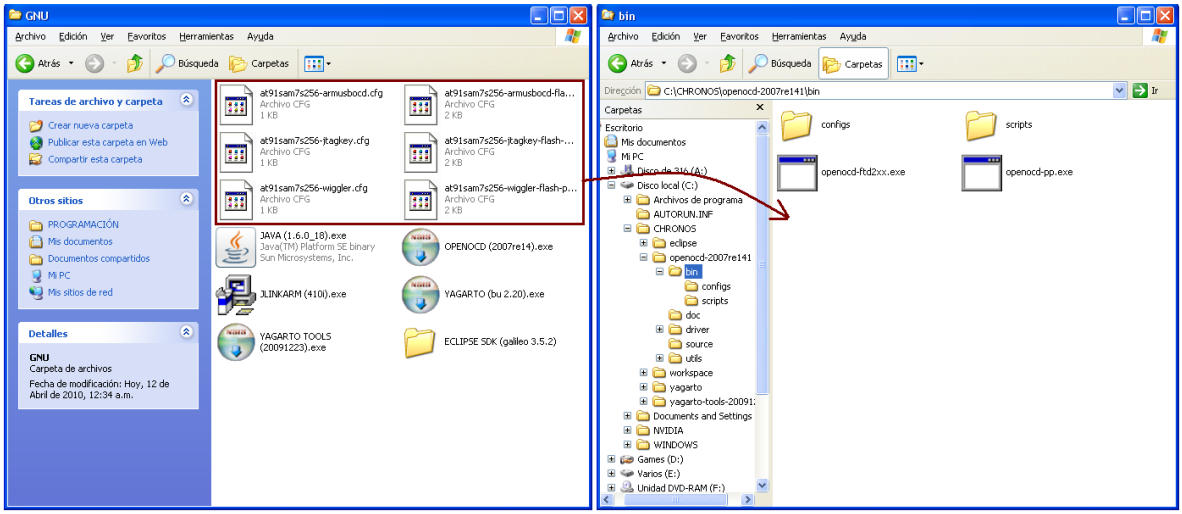


Figura 19: Pasando archivos .cfg a la carpeta bin de la instalación del openocd

→ Estando en Eclipse ir al menú *Run* -> external tools -> external tools configuration

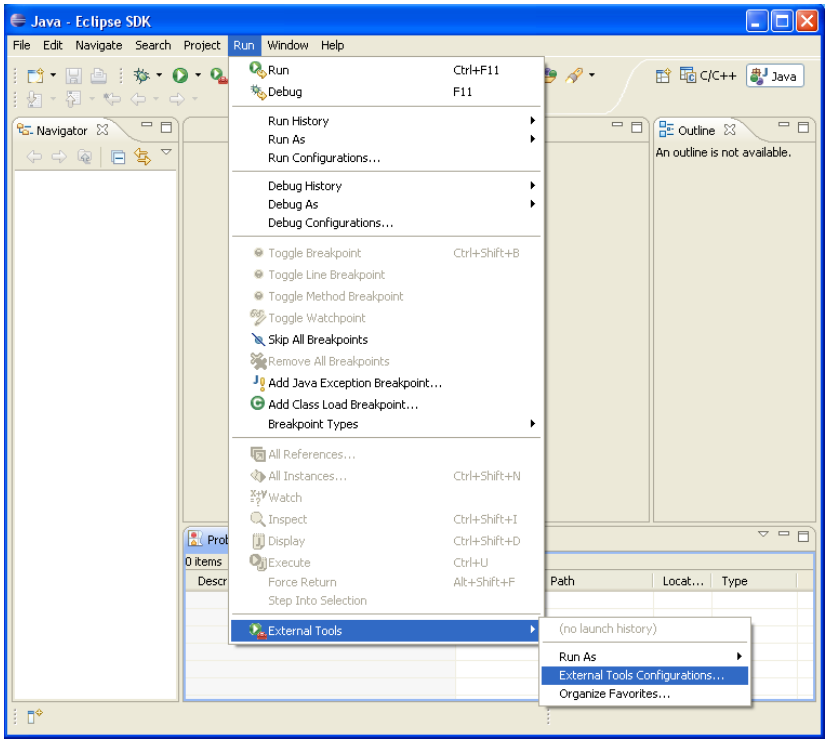


Figura 20: Seleccionando external tools configuration en eclipse

→ Dar click en *program* y luego en *new launch configuration* como se ve en la figura 21.

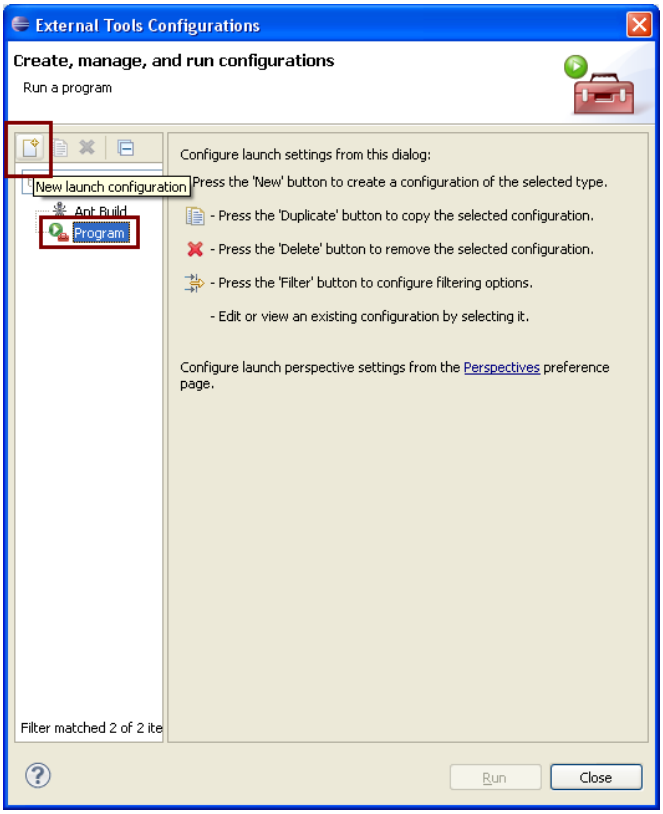


Figura 21: Ventana external tools configurations de eclipse

→ Aparecen algunos campos en la parte derecha de la ventana, en Name escribir OPENOCD. En Location seleccionar *browse file system,* y buscar en la ventana que se abre la ubicación donde se instaló el OPENOCD y entrar en la carpeta *bin* del mismo, allí seleccionar y abrir uno de los siguientes archivos según sea el caso:

- *openocd-ftd2xx.exe* [Para conexión via JTAG ICE (puerto paralelo a USB)]
- *openocd-pp.exe* [Para conexión via JTAG Wiggler (puerto paralelo)]

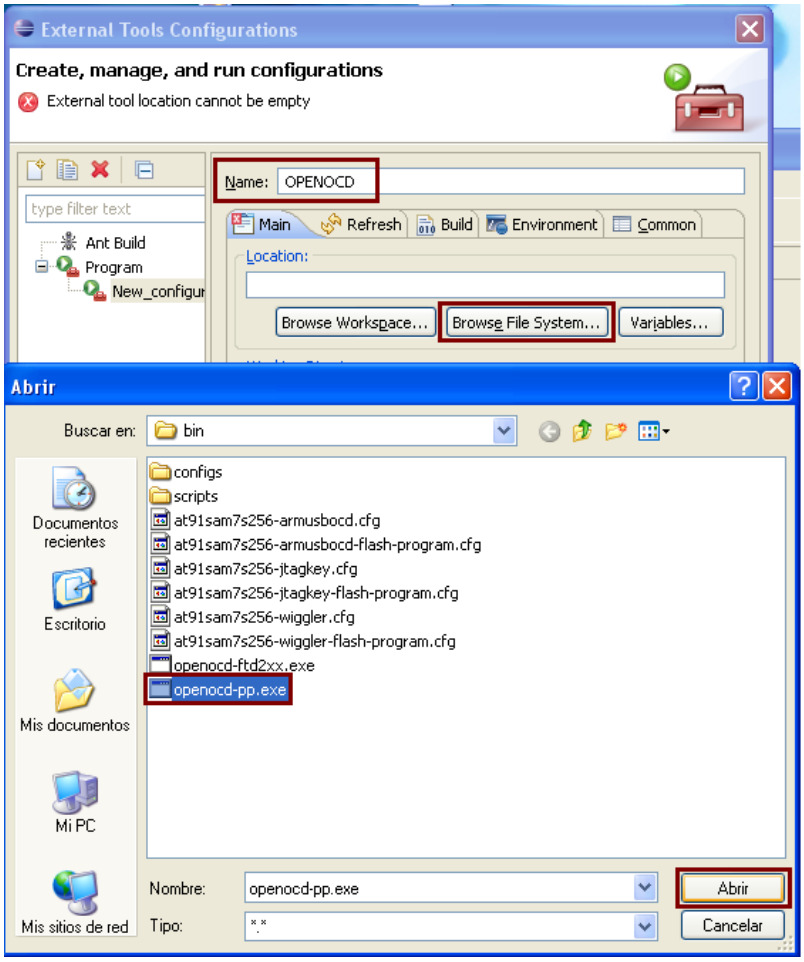


Figura 22: Abriendo el openocd para puerto paralelo

→ En *working directory* clickear en *browse file system* y allí buscar la carpeta bin que se encuentra en el directorio de instalación del OPENOCD, y dar click en aceptar.

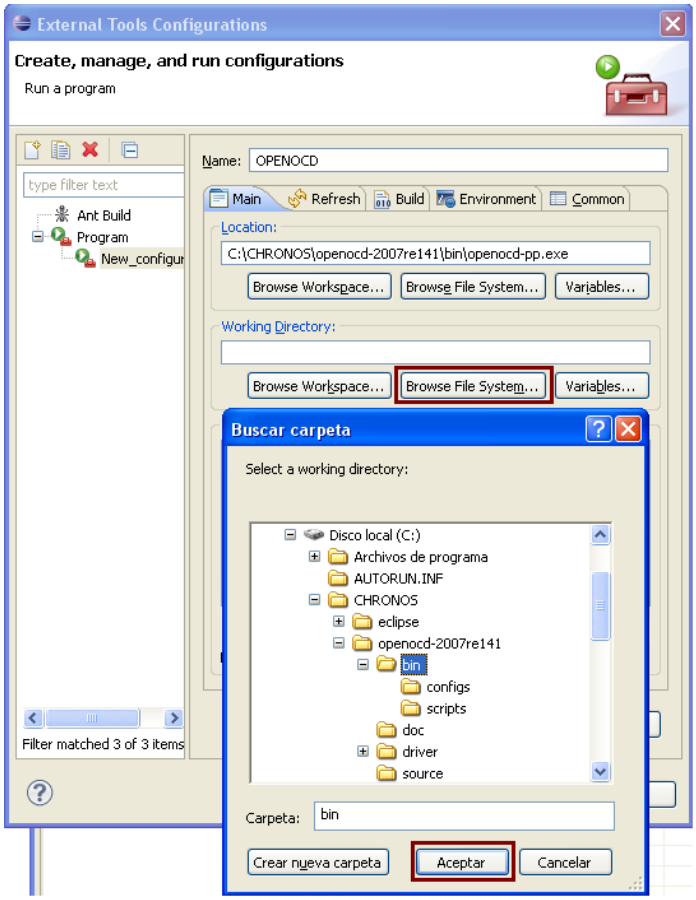


Figura 23: Seleccionando el working directory del openocd en eclipse

→ En Arguments escribir: *-f at91sam7s256-wiggler.cfg* tal y como se observa en la figura 24, luego seleccionamos *apply* y por último *close*.

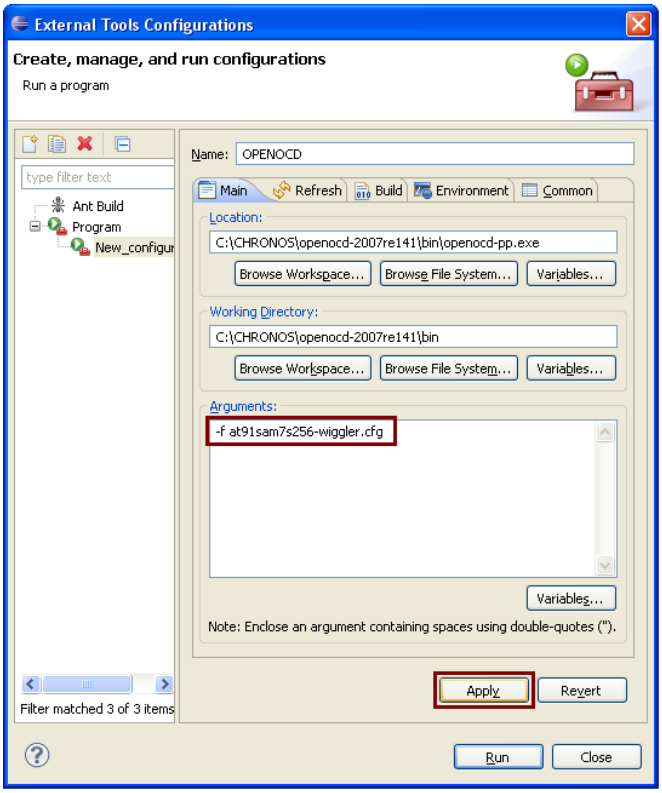


Figura 24: Escribiendo los argumentos necesarios para configurar el openocd en eclipse

5.3) Configurando C/C++ como perspectiva (si es necesario) [Para poder programar en C/C++]**:**

→ En Eclipse ir al menú *window -> open perspective -> other* y allí seleccionar C/C++ para por último dar click en *ok.*

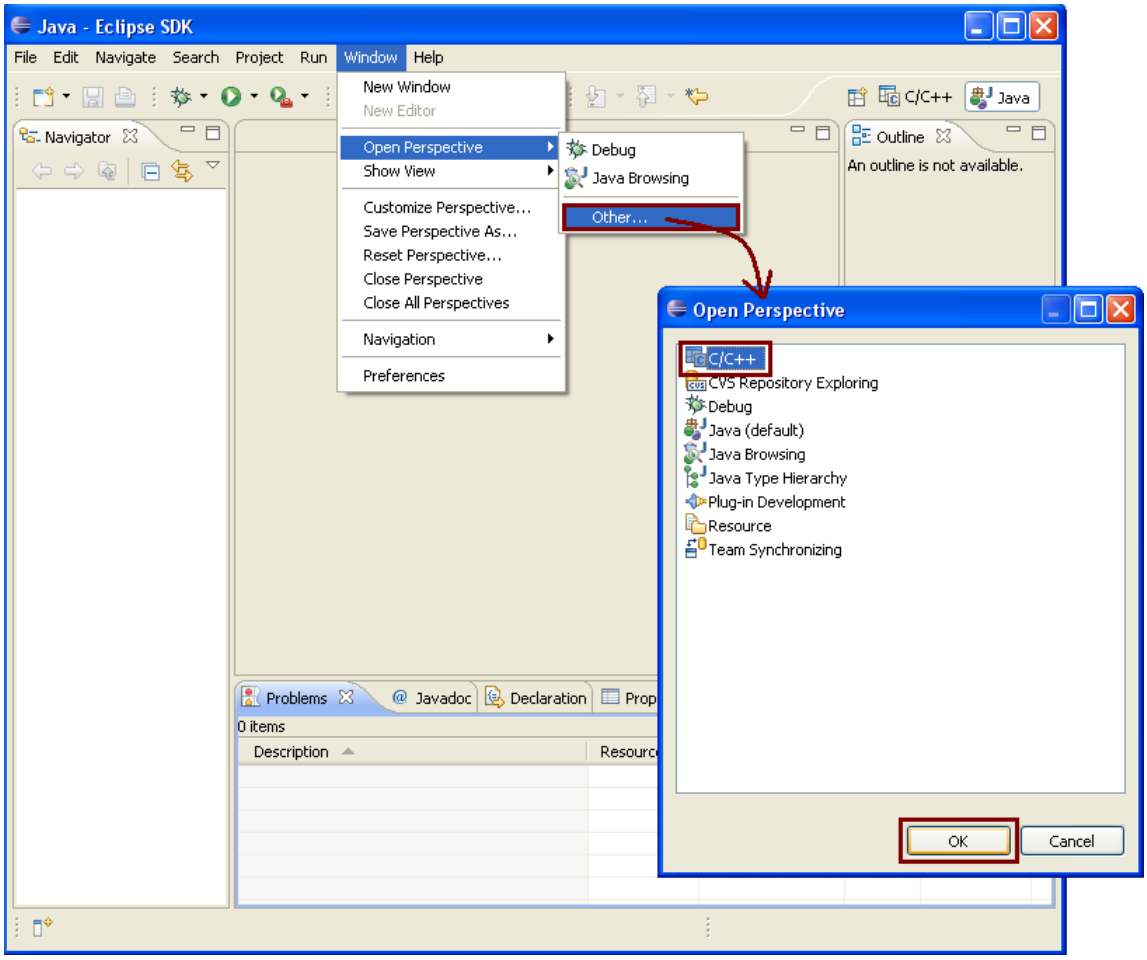


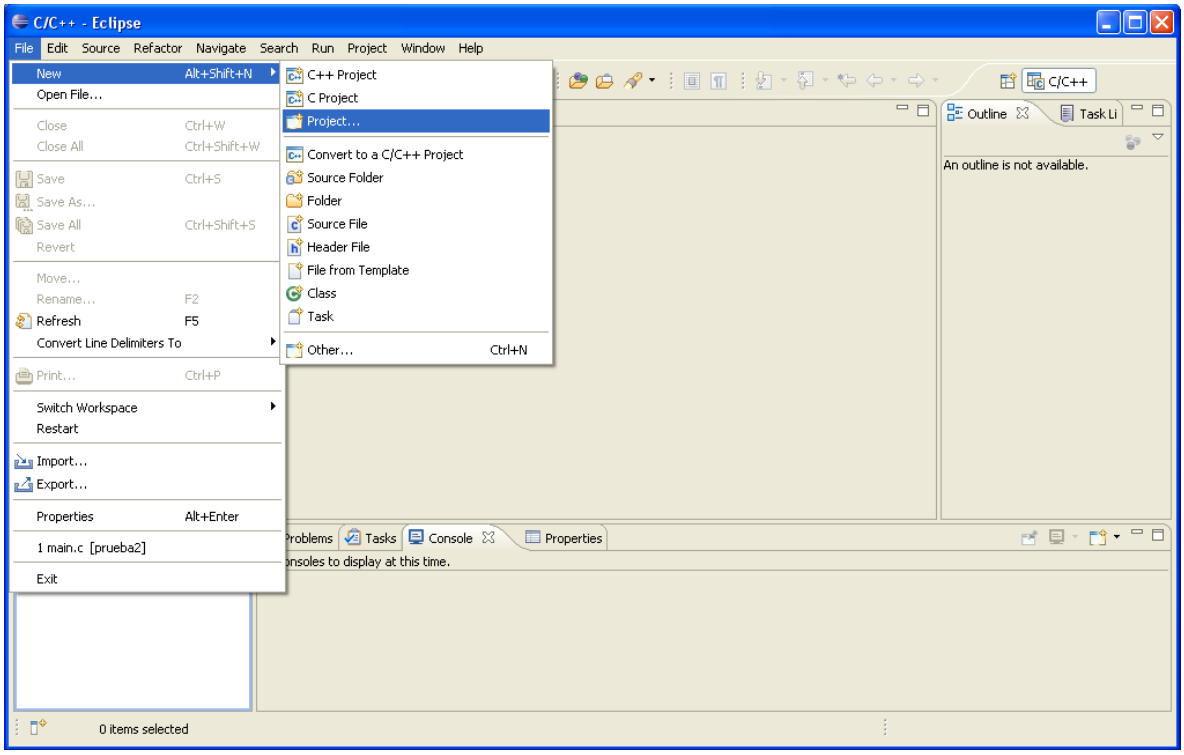
Figura 25: Configurando C/C++ como perspectiva en eclipse

CHRONOS

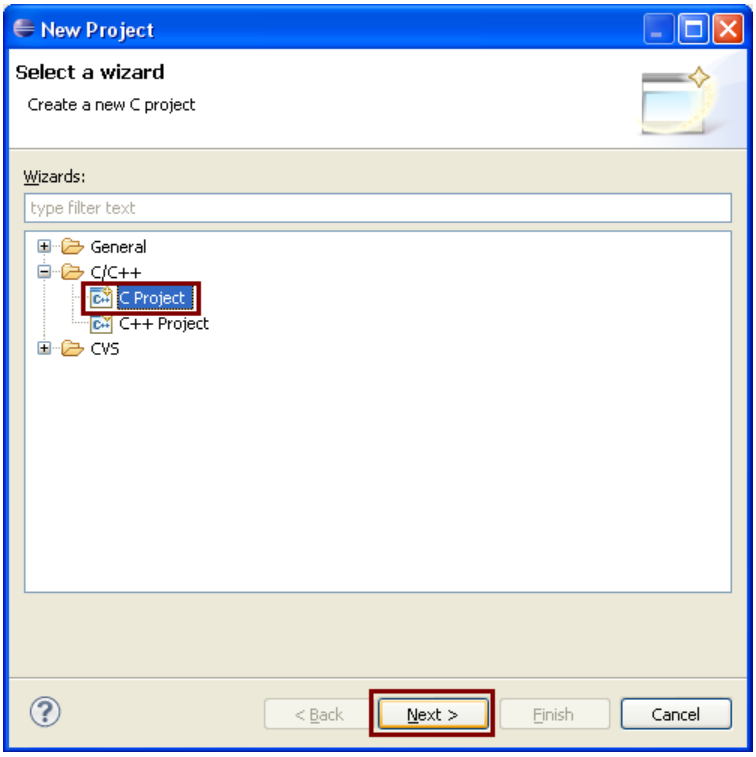
PROGRAMANDO EN ECLIPSE

1) Creando un proyecto nuevo

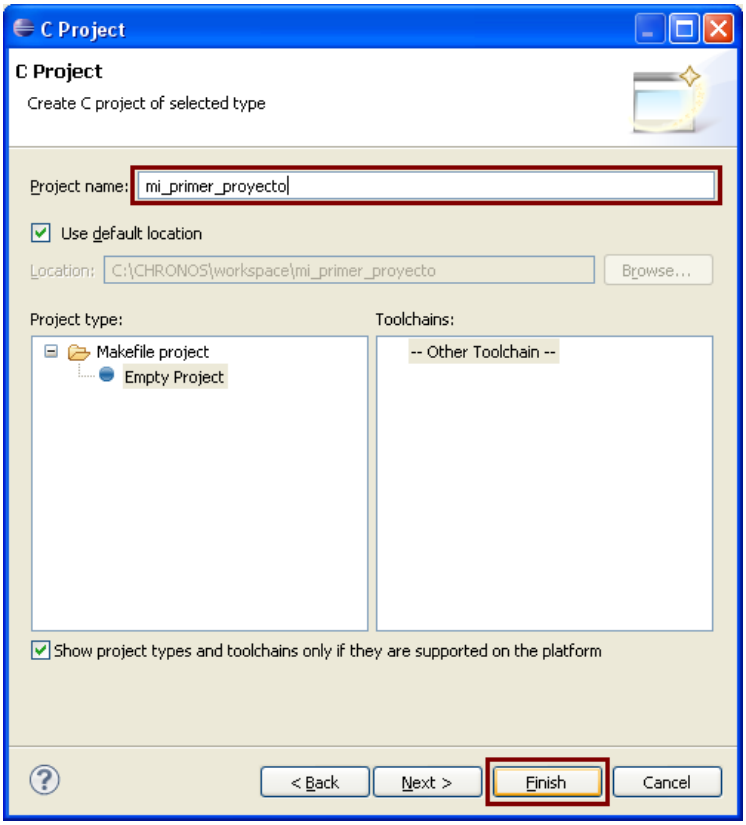
Estando en el workbench de Eclipse ir a *File -> New -> Project….*



Seleccionar *C project* y luego click en *next.*

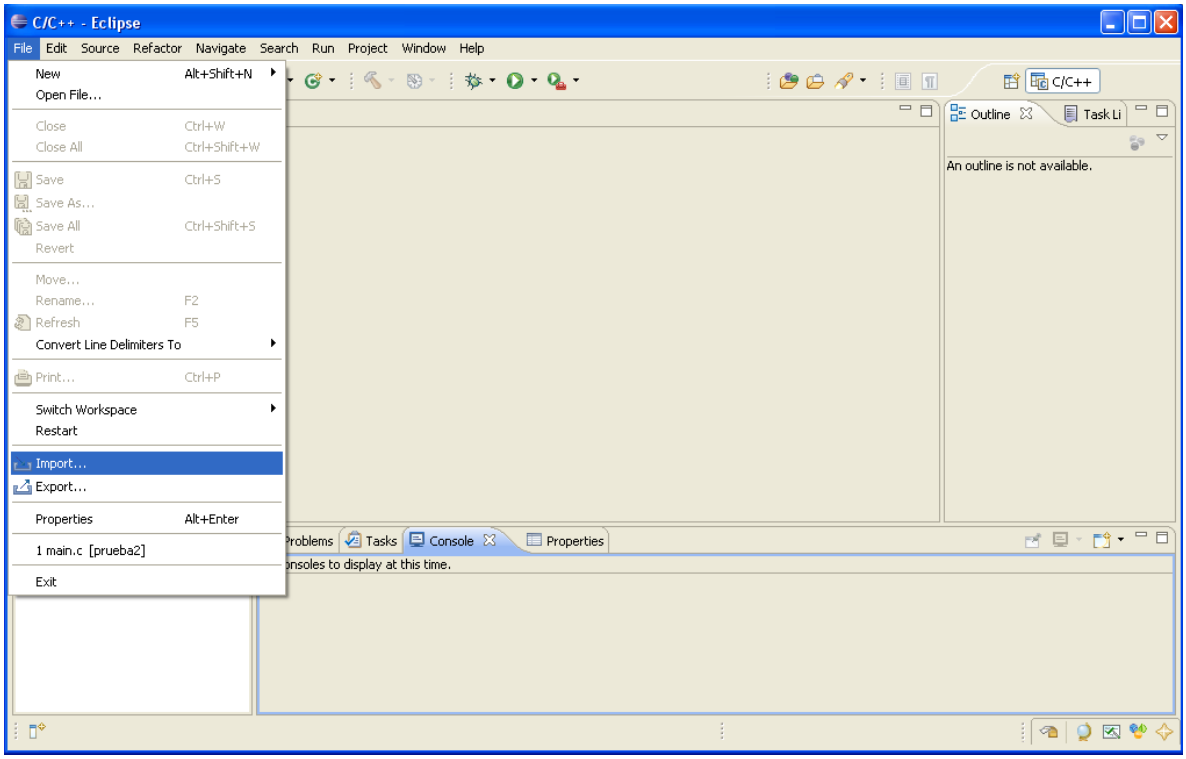


Luego colocarle un nombre al nuevo proyecto, y configurar las opciones tal y como se muestra en la imagen para luego seleccionar *finish*.

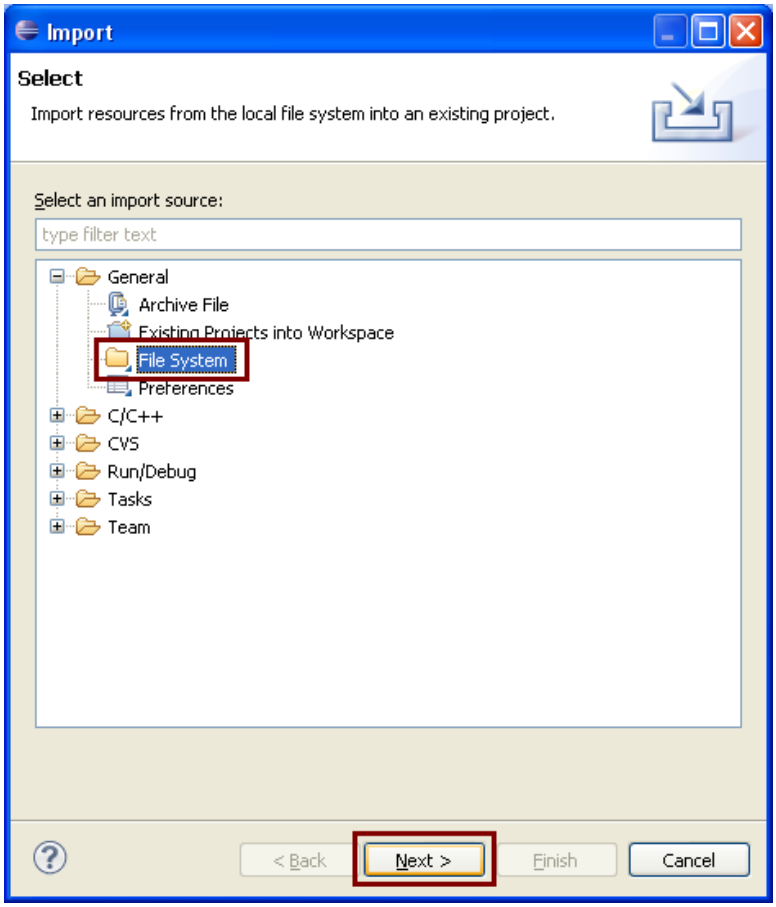


Esperamos un momento y luego volveremos al Workbench donde en el panel de la izquierda (en el *Project explorer*) encontraremos el proyecto que creamos.

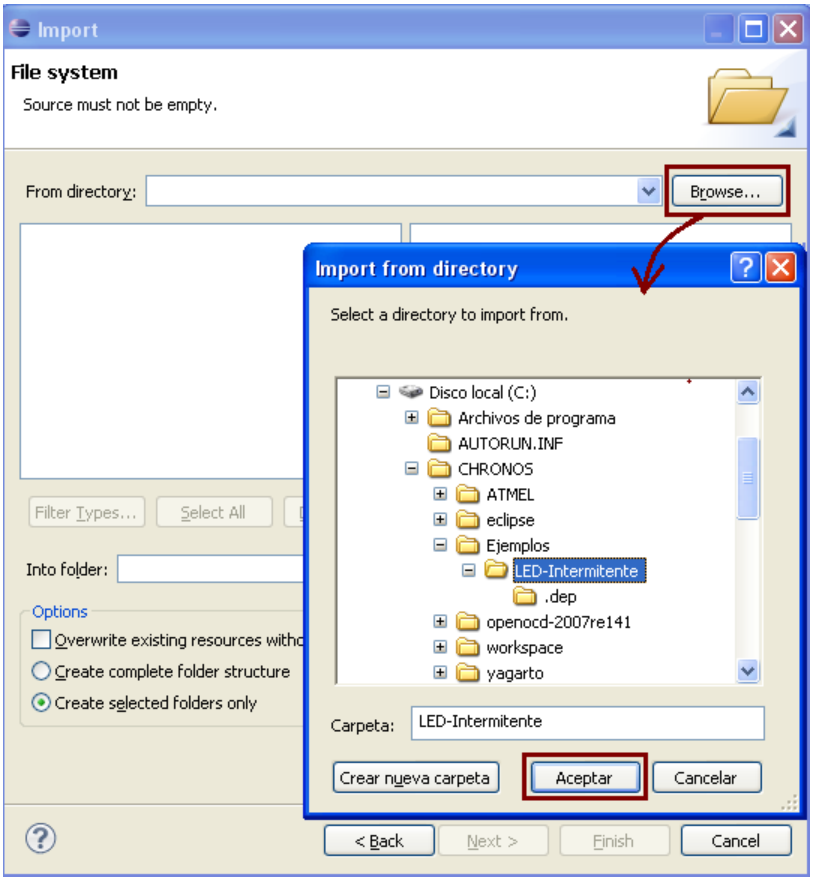
2) **Importando un proyecto ya creado** [para abrir proyectos ya existentes]**:**
En la ventana principal de Eclipse (workbench) ir a *File -> Import…*



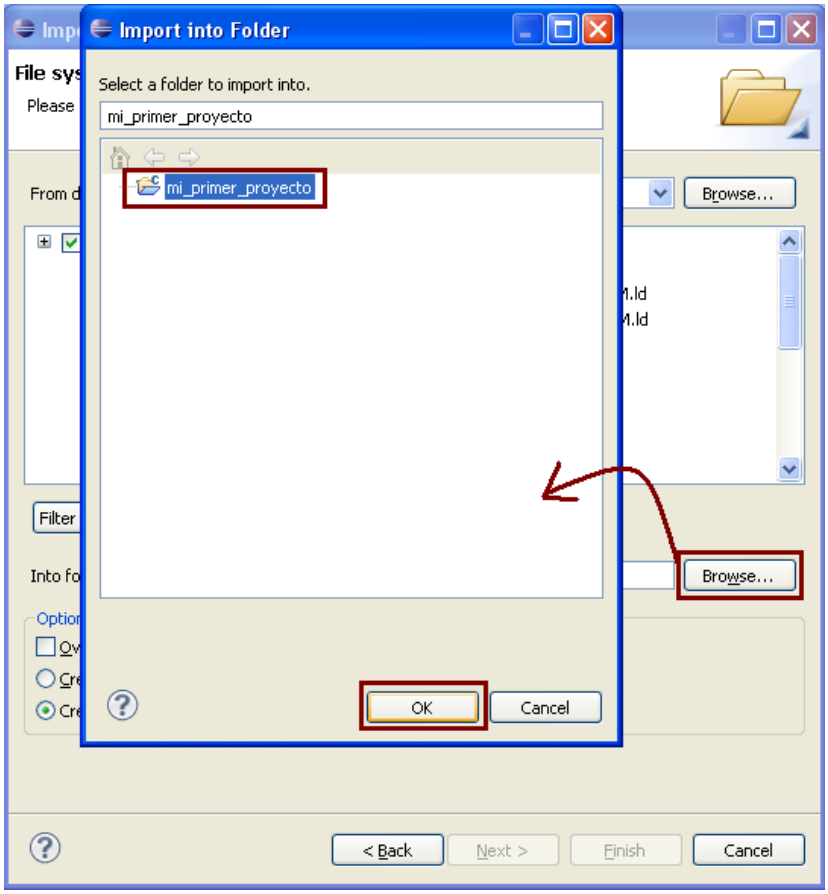
Luego, en la ventana que se abre seleccionar *file system* y luego click en *next*, tal y como la imagen lo indica.



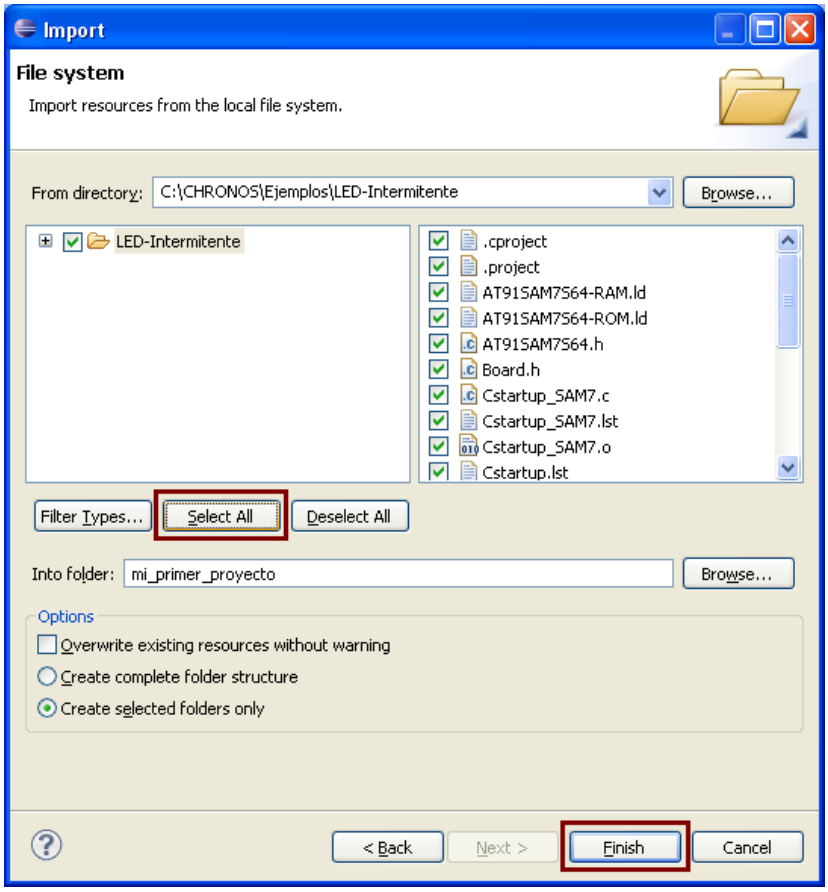
Al frente de *from directory,* Dar click en *browse* (para seleccionar el origen) para buscar la carpeta que contiene los archivos del proyecto que se desea importar, tal y como se presenta en la imagen siguiente.



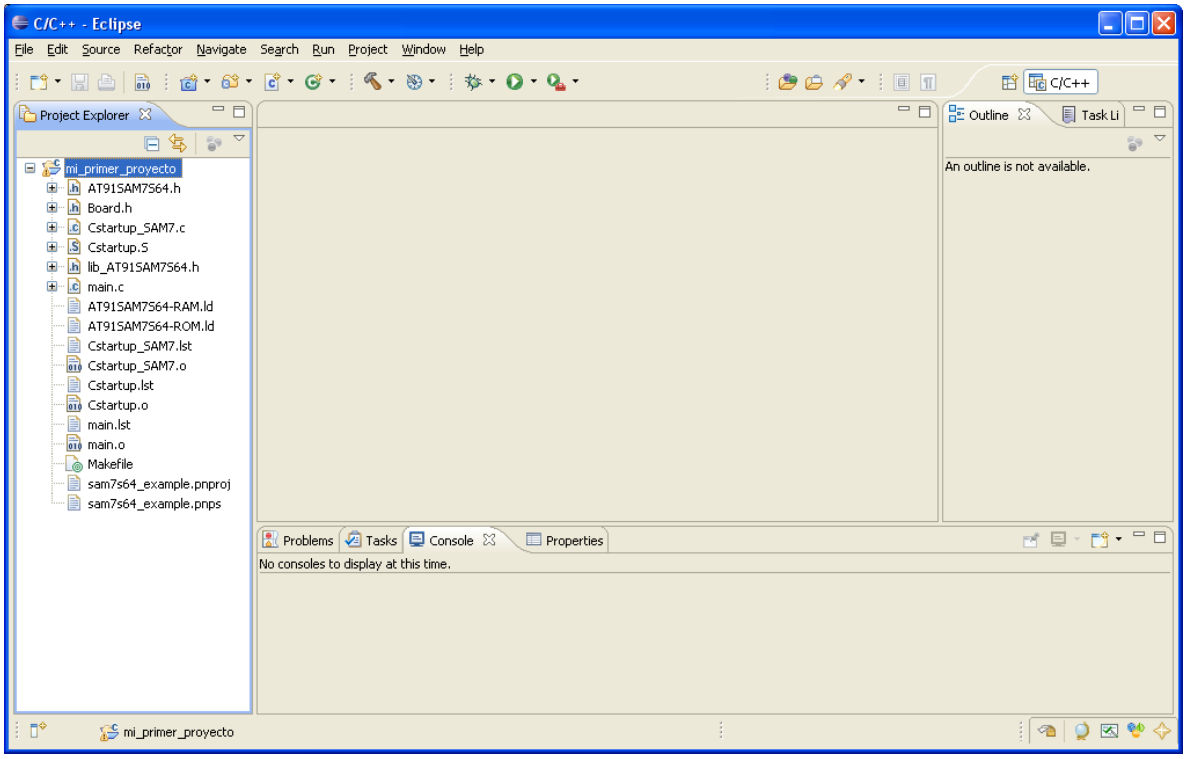
Así mismo, al frente de *into folder* dar click en *browse* (para seleccionar el destino del proyecto)



Luego dar click en *finish* verificando las opciones tal y como se muestran en la siguiente figura.

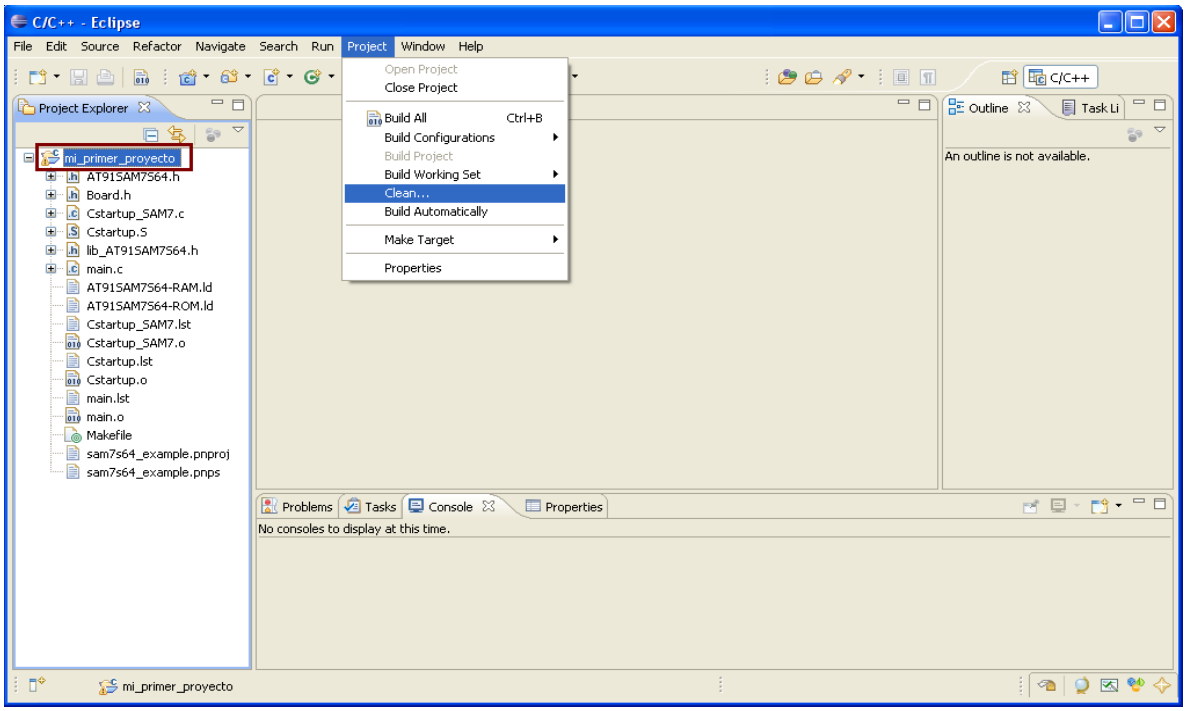


Seguidamente en la ventana que nos pregunta si queremos sobrescribir el proyecto optamos por *no to all*, finalmente tendremos un workbench donde podremos expandir nuestro proyecto y observar todos los archivos que contiene.

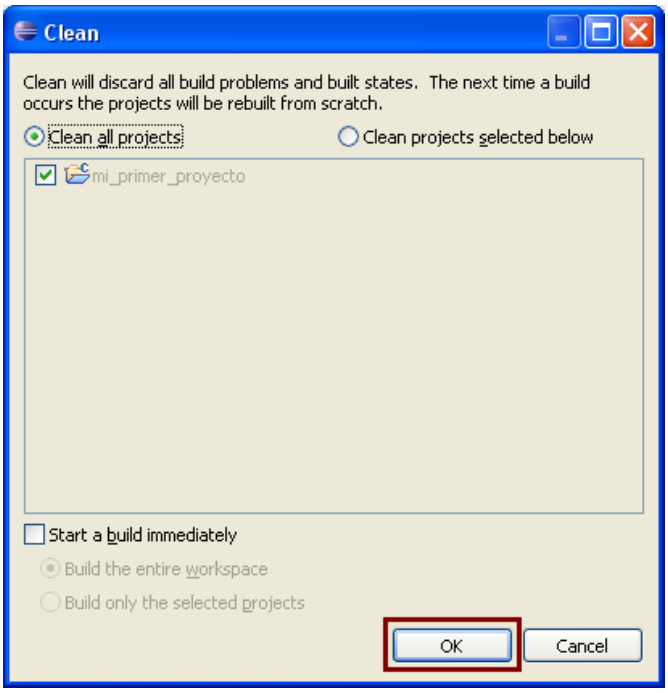


3) Limpiando el proyecto [Para eliminar archivos antiguos que pueden afectar la correcta construcción del proyecto]**:**

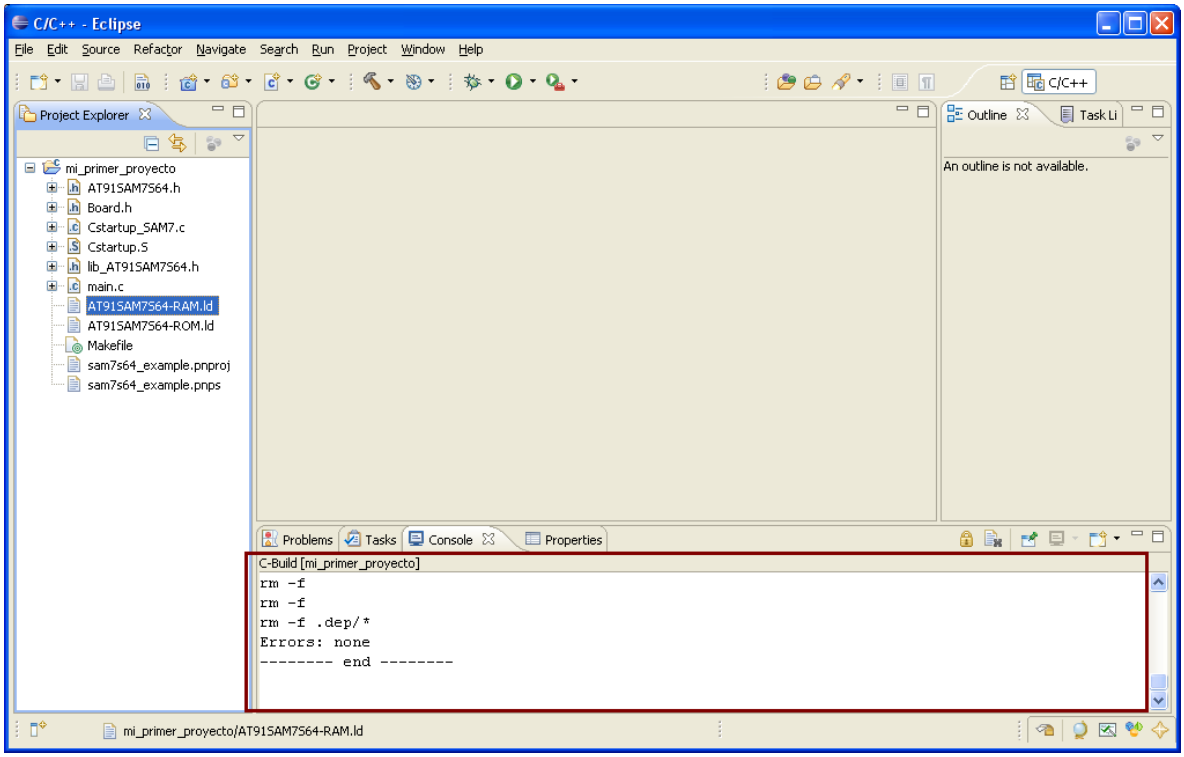
Primero dar click sobre el proyecto a limpiar (que se encuentra en el explorador de proyectos en el panel izquierdo de Eclipse). Luego, ir al menú *project -> clean…*



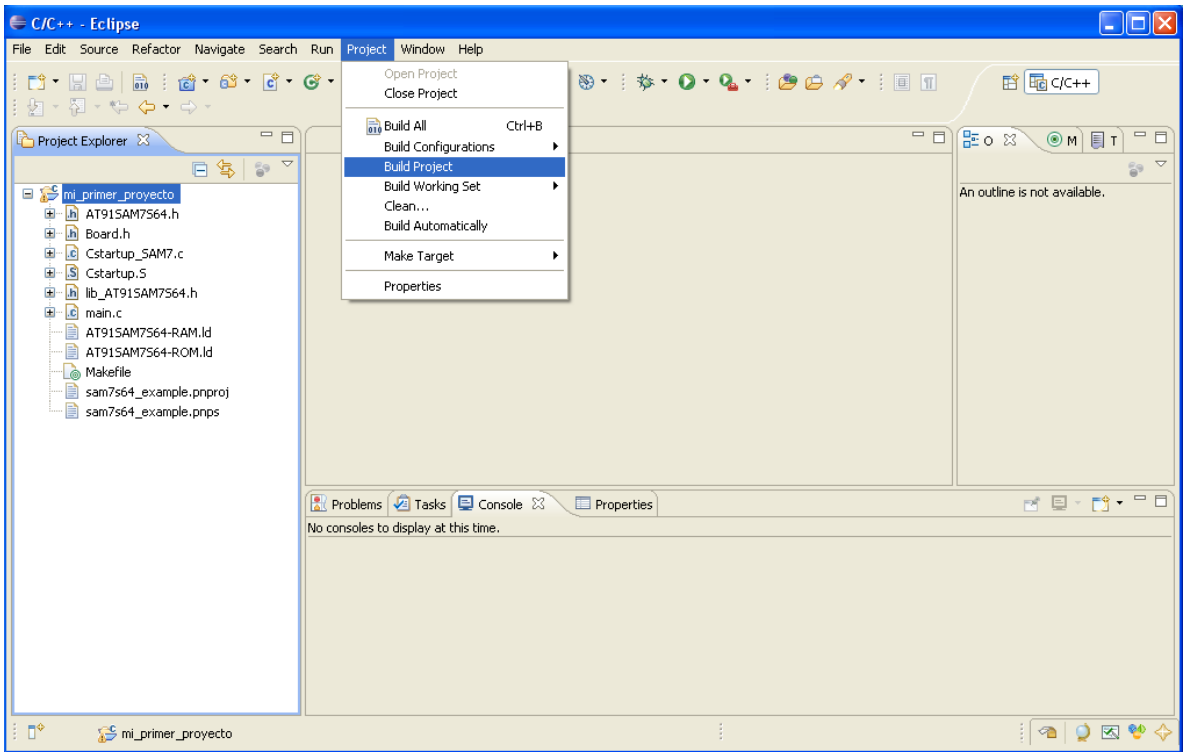
En la ventana que se abre se elijen las opciones tal y como se muestra en la siguiente figura.



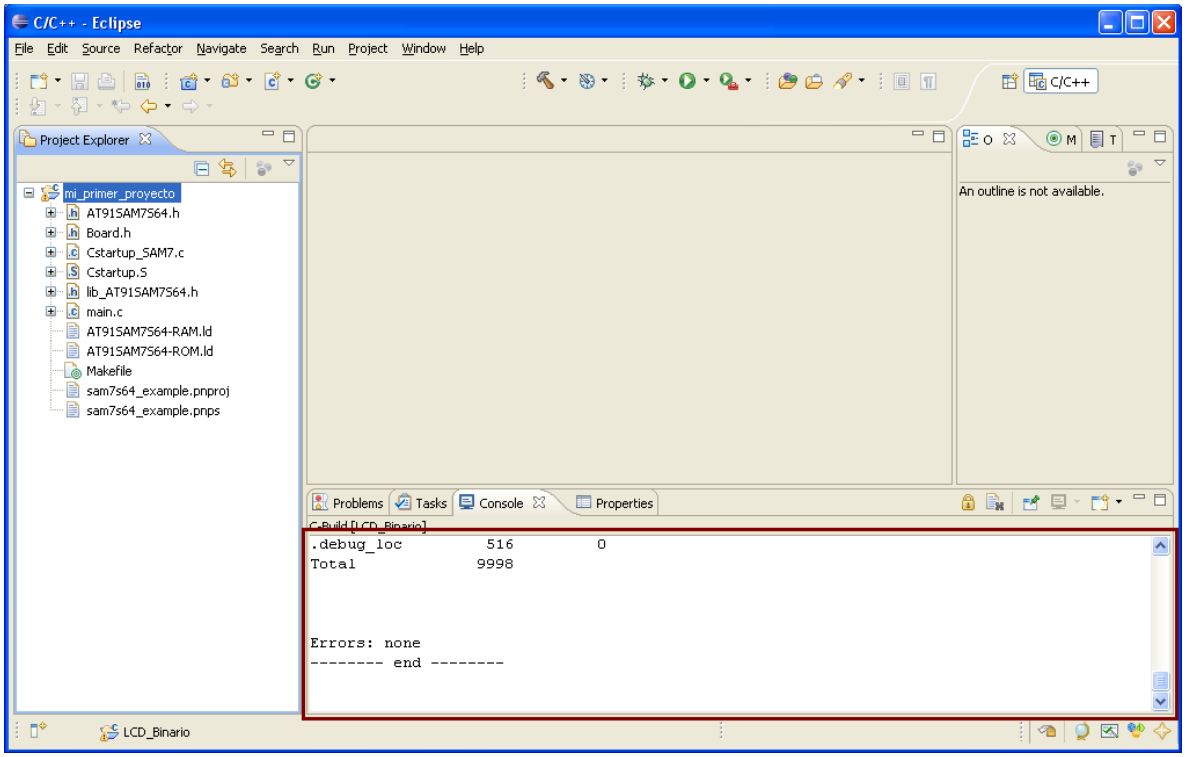
Si el proceso de limpieza del proyecto se llevo a cabo correctamente, en la pestaña *console* que está ubicada en la parte inferior de Eclipse debe aparecer que no hubo errores (*errors: none*) tal y como se muestra en la siguiente imagen.



4) **Construyendo el proyecto** [Proceso en el que se compila el código y se construye el proyecto] Luego de dar click sobre el proyecto a construir (que se encuentra en el explorador de proyectos en el panel izquierdo de Eclipse) ir a menú *project ->Build proyect…*

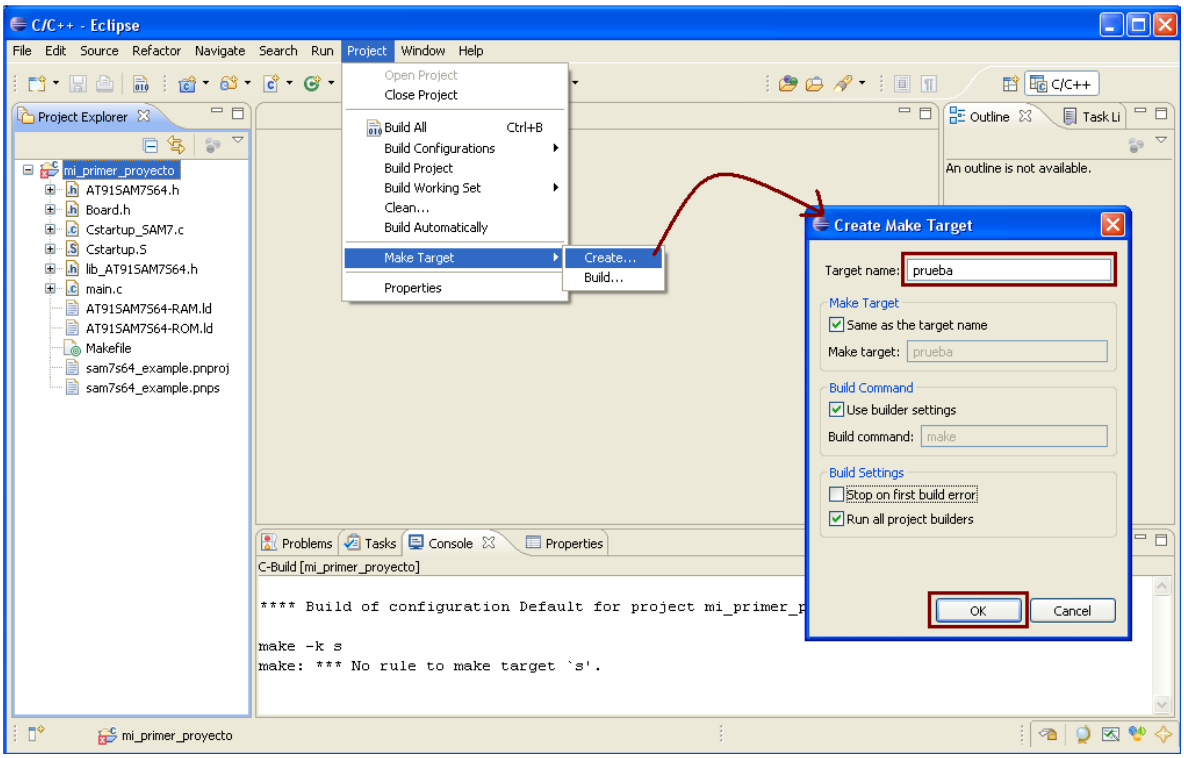


Luego de esperar a que la barra de progreso este completa, y si se construyó con éxito el programa, al igual que en el caso anterior, en la consola debe aparecer *Errors none*, tal y como se aprecia en la siguiente imagen.



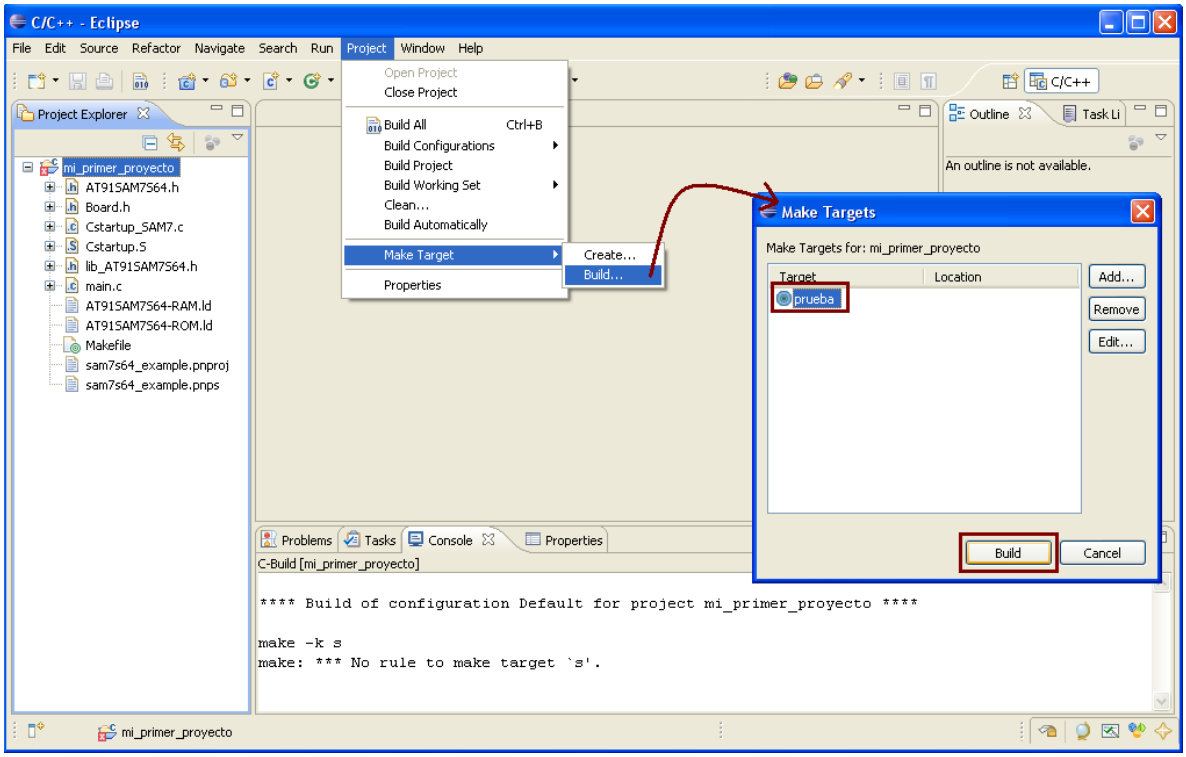
En este punto, Eclipse ha generado todos los archivos que configuran y constituyen el proyecto entre estos se encuentra el archivo binario, que se reconoce porque después del nombre (generalmente *"main"*) le sigue *.bin*. Dado a que el archivo binario es el que contiene el programa en lenguaje de máquina, este puede ser enviado al microcontrolador haciendo uso de otros medios y programas, tal es el caso del SAM-BA, que permite enviar el archivo binario (programar el microcontrolador) a través del puerto USB (para el manejo de esta herramienta remitirse al tutorial de SAM-BA).

5) "Haciendo" el proyecto [Proceso mediante el cual se programa el microcontrolador]**:**
Seleccionamos *project -> make target -> create…*



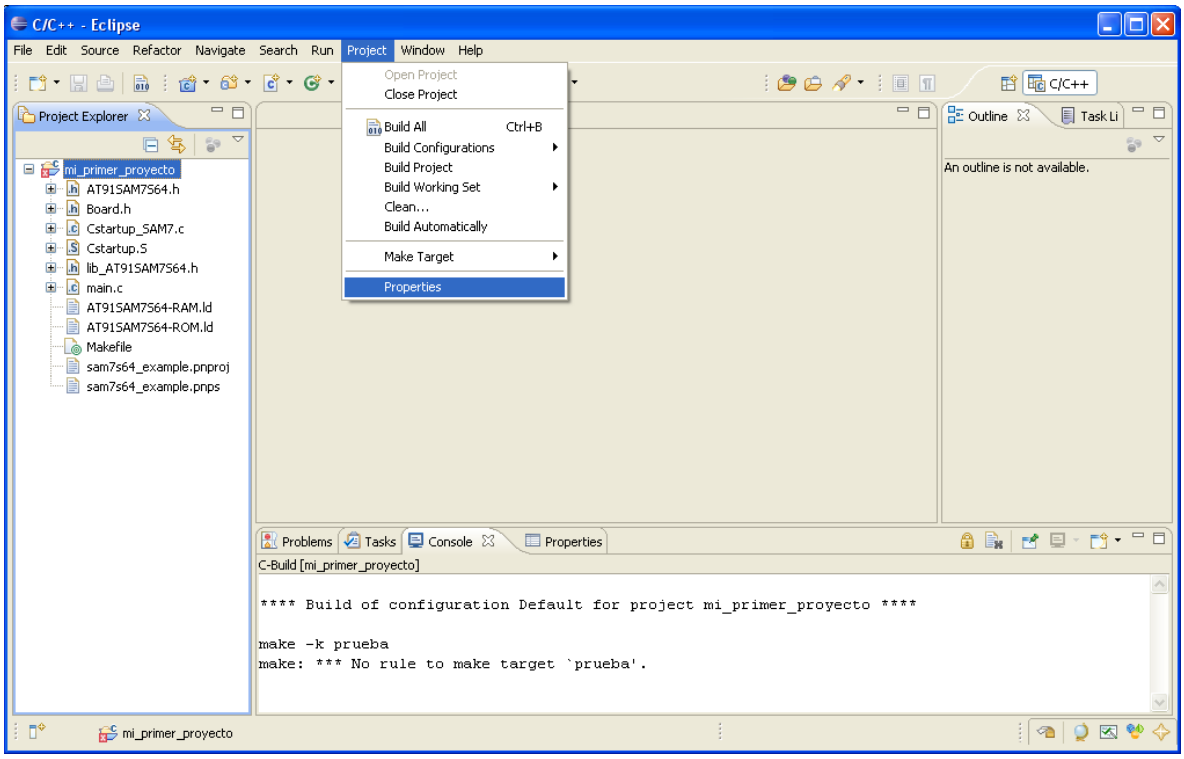
Allí colocamos un nombre cualquiera al proceso, y marcamos las opciones tal y como se observó en la figura anterior, para posteriormente clickear en *ok*.

Luego de volver al workbench e ir al menú project -> make target -> build… en la ventana que aparece seleccionar el nombre del proceso que creamos anteriormente y posteriormente el dar click en build así:

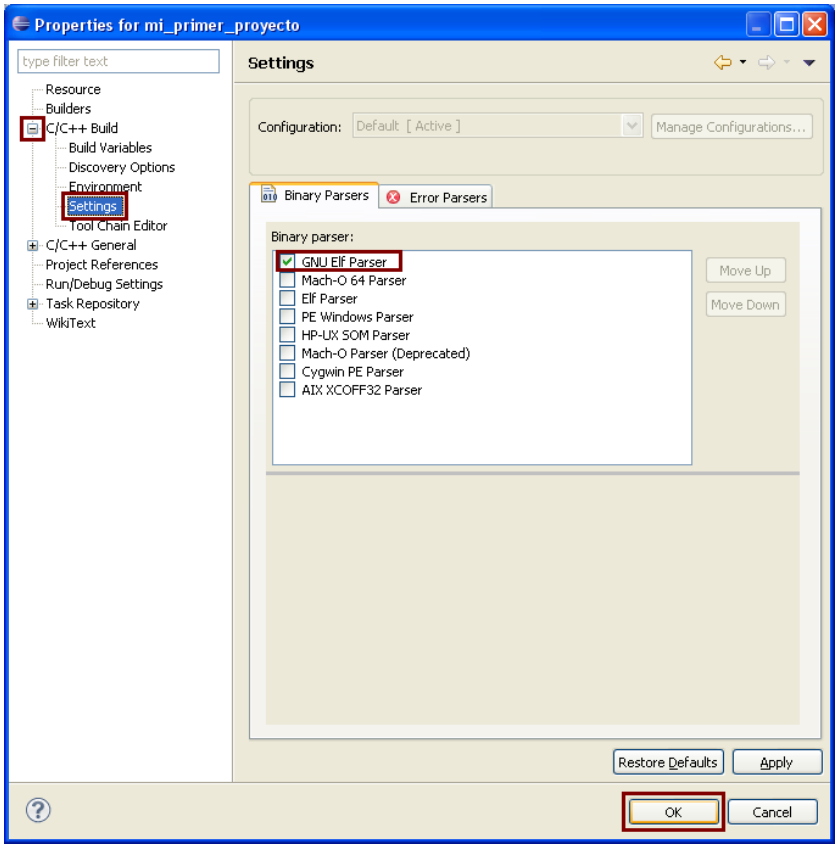


Si el proceso se llevo a cabo correctamente aparecerá en la consola que no hubo errores y el microcontrolador habrá quedado programado y listo para ejecutar el programa.

Es necesario configurar el *parser* (analizador) indicado para el desarrollo de nuestros proyectos, para ello en el wokbench de Eclipse vamos a project -> properties



Luego en la ventana que nos aparece, vamos al panel izquierdo y abrimos el desplegable *C/C++ Builder* para luego seleccionar *settings,* de las opciones que aparecen en el espacio derecho de la ventana seleccionamos *GNU elf parser*. Por último click en *ok.*


Properties for mi_primer_proyecto
type filter text
Resource
Builders
C/C++ Build
Build Variables
Discovery Options
Environment
Settings
Tool Chain Editor
C/C++ General
Project References
Run/Debug Settings
Task Repository
WikiText
Settings
Configuration:
Default [Active]
Manage Configurations...
Binary Parsers
Error Parsers
Binary parser:
GNU Elf Parser
Mach-O 64 Parser
Elf Parser
PE Windows Parser
HP-UX SOM Parser
Mach-O Parser (Deprecated)
Cygwin PE Parser
AIX XCOFF32 Parser
Move Up
Move Down
Restore Defaults
Apply
OK
Cancel

SAM-BA

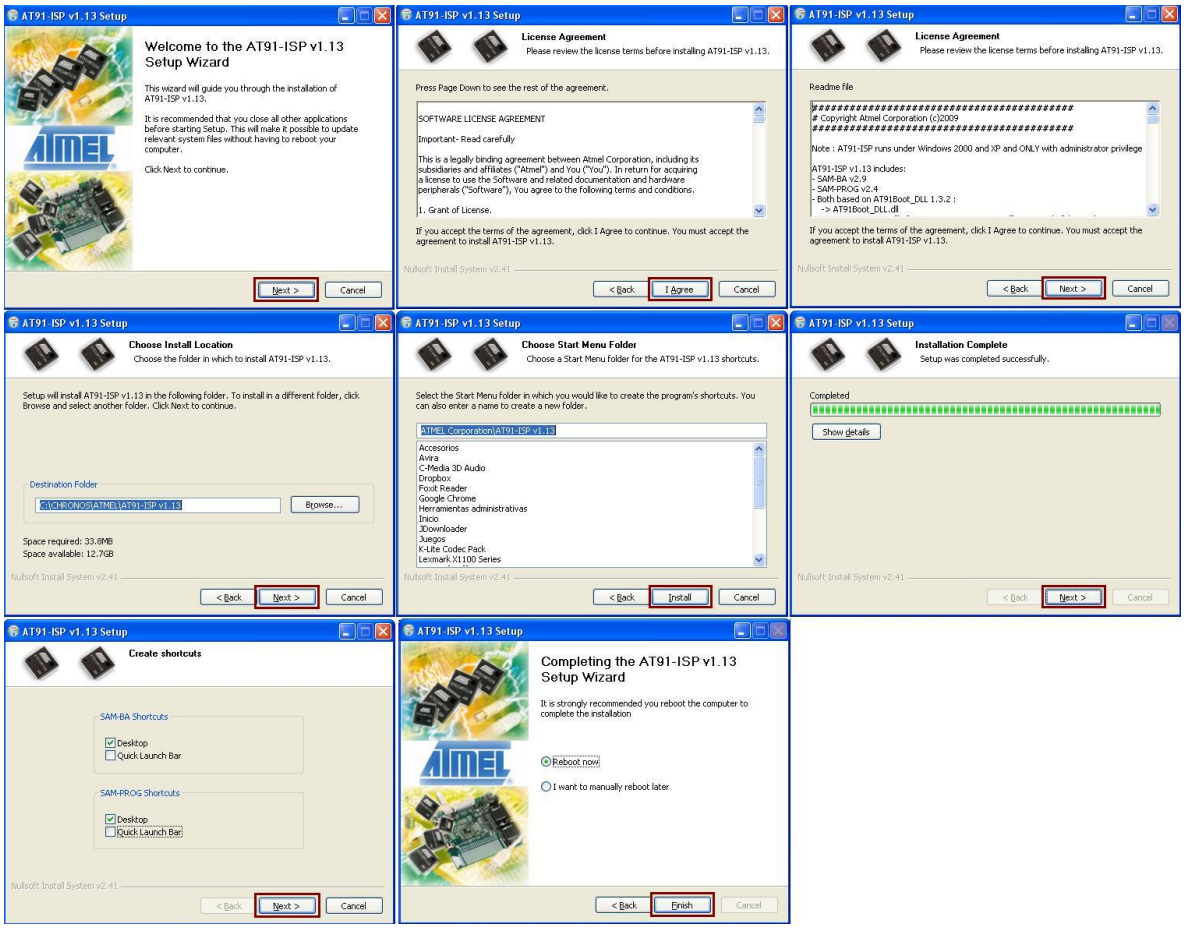
SAM-Boot Assistant Esta es una herramienta que cuenta con una interface para el usuario y que permite comunicar el PC con el microcontrolador a través del puerto **USB**, para esto el microcontrolador debe tener cargado el bootloader que para nuestro microcontrolador se llama también SAM-BA, También es necesario que la tarjeta base tenga conectado un cristal de 18.432 MHz, con otro tipo de cristal no se puede establecer la comunicación con el PC via USB usando el SAM-BA (toca utilizar otra manera de comunicación con el PC, como puede ser SAM-ICE, JTAG Wiggler, etc.)

Cargar el bootloader:

El procesos consiste simplemente en unir el pin TEST del microcontrolador a Vcc (3.3 V) durante 10 segundos, mientras este esté alimentado correctamente. En la tarjeta base consiste simplemente en conectarle la alimentación (sea USB o externa), colocar el jumper que dice TST, encenderla, esperar 10 segundos, apagarla quitarle el jumper y volverla a prender. La siguiente foto muestra la ubicación del jumper TST en la tarjeta base.

Instalación:

Luego, es necesario instalar el AT91 ISP (In-system Programing) en el PC, y para esto ejecutamos el instalador correspondiente y seguimos los pasos tal y como se observa en la siguiente imagen.



Luego de que el PC se halla reiniciado, conecta la tarjeta base (con el microcontrolador) al computador utilizando el cable USB, cuando lo hagas aparecerá el siguiente icono en la barra de notificaciones  . Seguidamente el asistente para nuevo hardware de Windows se abrirá, y en la barra de notificaciones aparecerá

Para la correcta instalación del dispositivo se deben seguir los pasos que se indican en la siguiente imagen, donde lo que hacemos es decirle al PC que busque el controlador adecuado para la tarjeta base, y que lo instale.

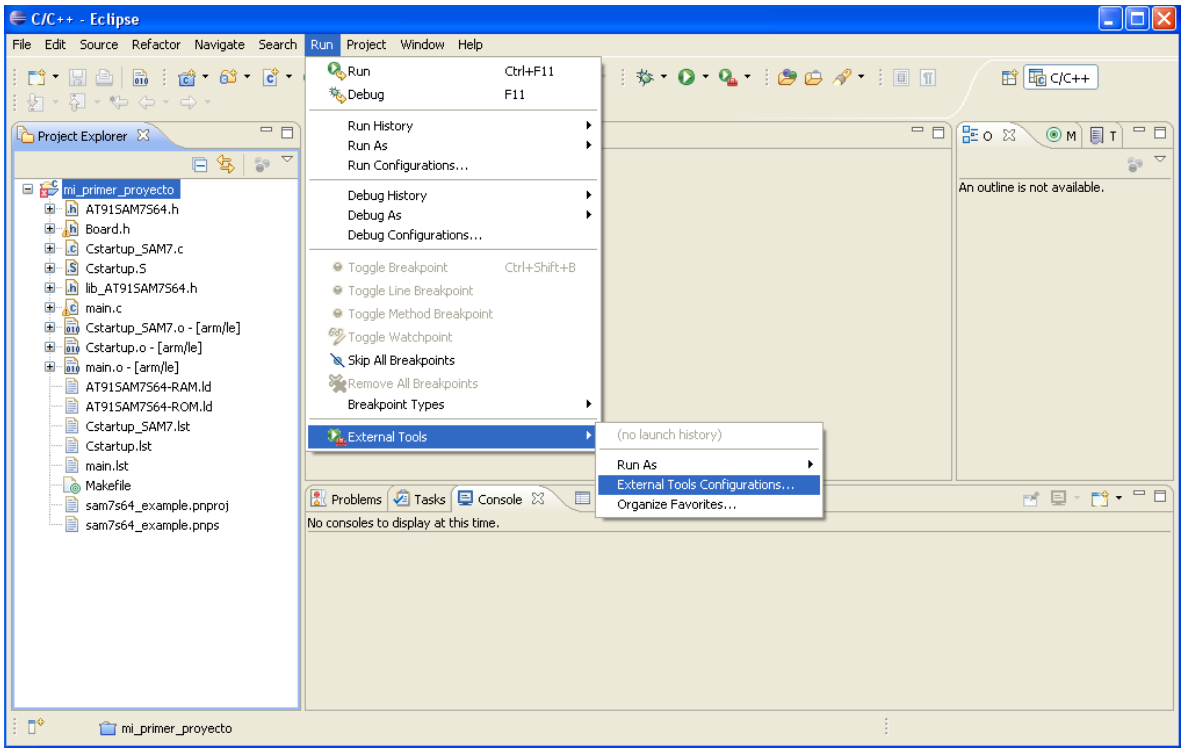
Luego de terminado el proceso, y si todo salió bien, el mensaje en la barra de notificaciones ahora debe decir

Y de ahora en adelante, cada vez que la tarjeta base con el microcontrolador sea conectada y reconocida correctamente por el PC, aparecerá el icono  y podemos proceder a enviar el archivo .bin del programa que queremos que el microcontrolador ejecute.

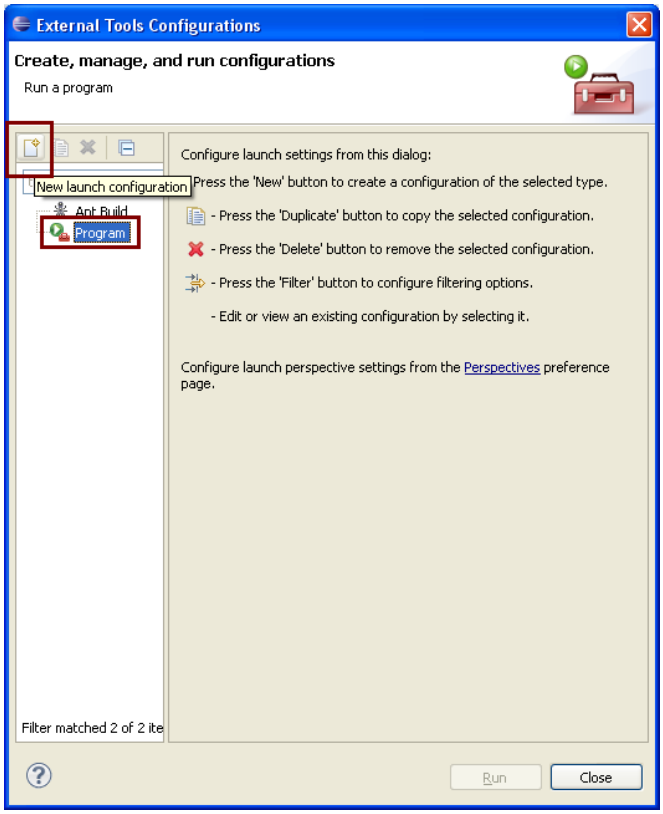
Configuración en Eclipse:

Proceso importante si queremos integrar todas las herramientas de desarrollo a nuestro IDE, y así trabajar de manera más ordenada y eficiente.

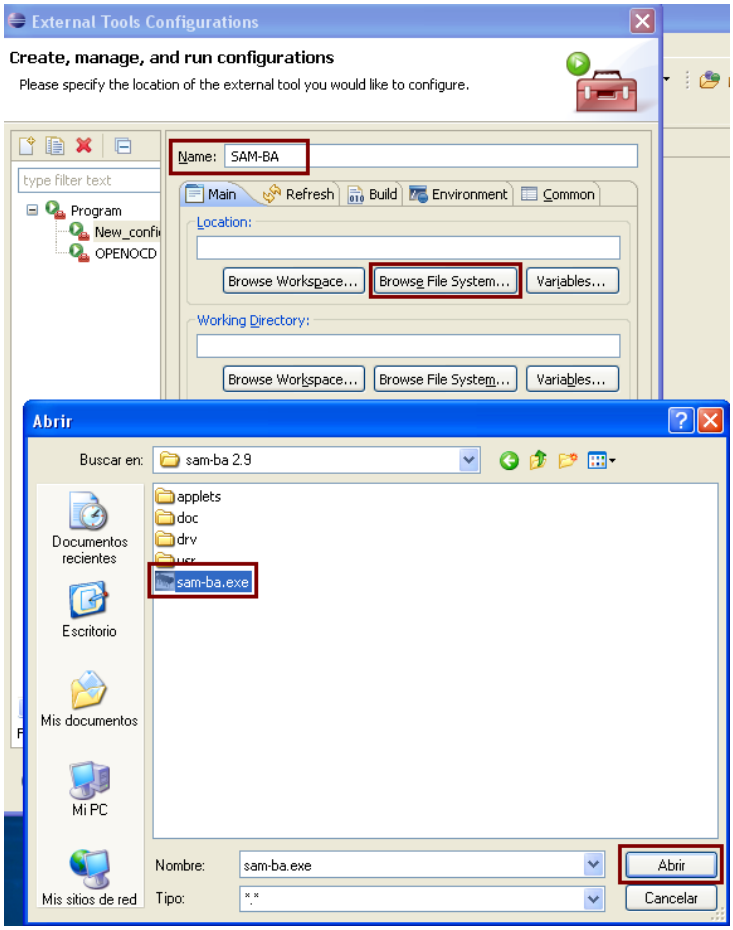
Hacer click en *Run -> External tools -> External tools configurations…*



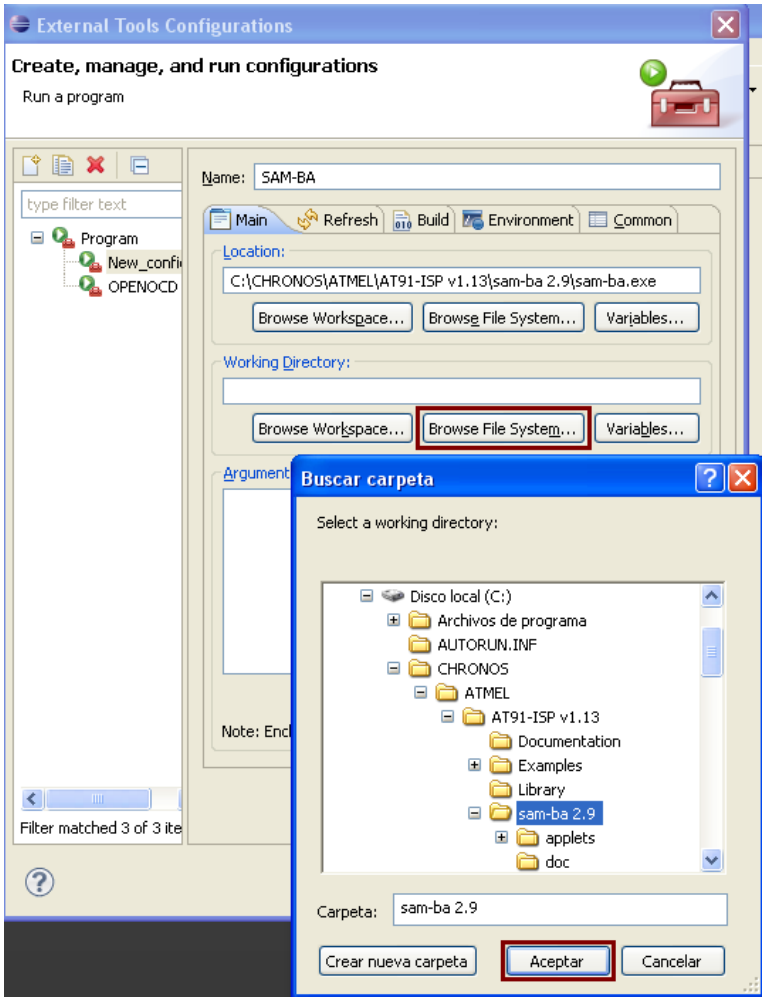
Aparecerá la ventana de *External Tools Configurations*, allí dar click en *Program* y luego pulsar el botón *New*, tal y como se ve en la siguiente imagen.



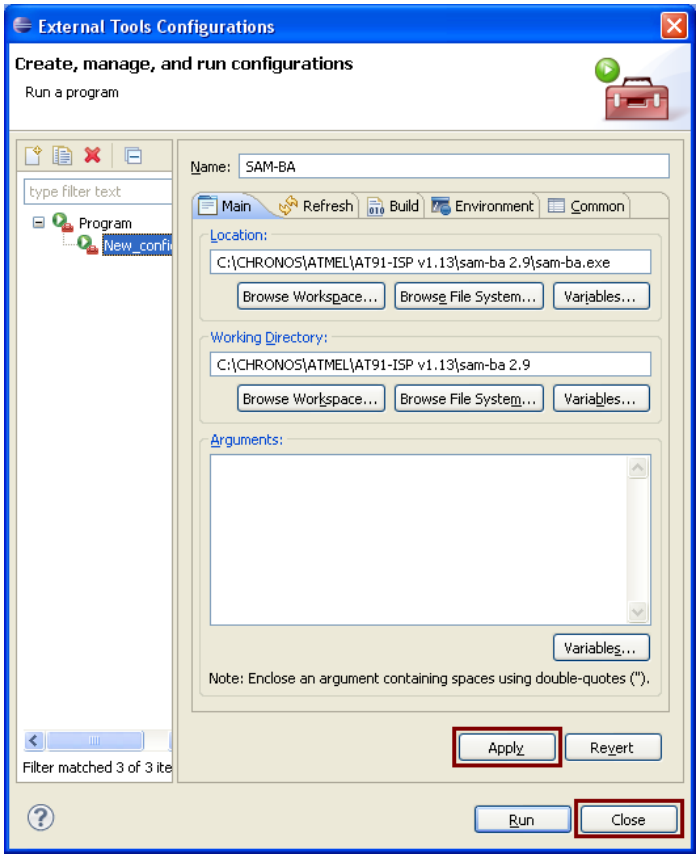
Aparecen algunos campos en la parte derecha de la ventana, en Name escribir SAM-BA. En *Location* seleccionar *Browse File System..,* buscar en la ventana que se abre la ubicación donde se instaló el AT91 ISP, y buscar y seleccionar en la carpeta *sam-ba 2.9* el archivo .exe que se llama *sam-ba.exe*, como se aprecia a continuación.



Luego en el campo de *working directory* colocar la ruta de la carpeta principal del sam-ba, para esto clickear en el botón *Browse File System…* y en la ventana que se abre ubicar y seleccionar la carpeta sam-ba 2.9 que se encuentra en la carpeta donde se instaló el AT91 ISP.

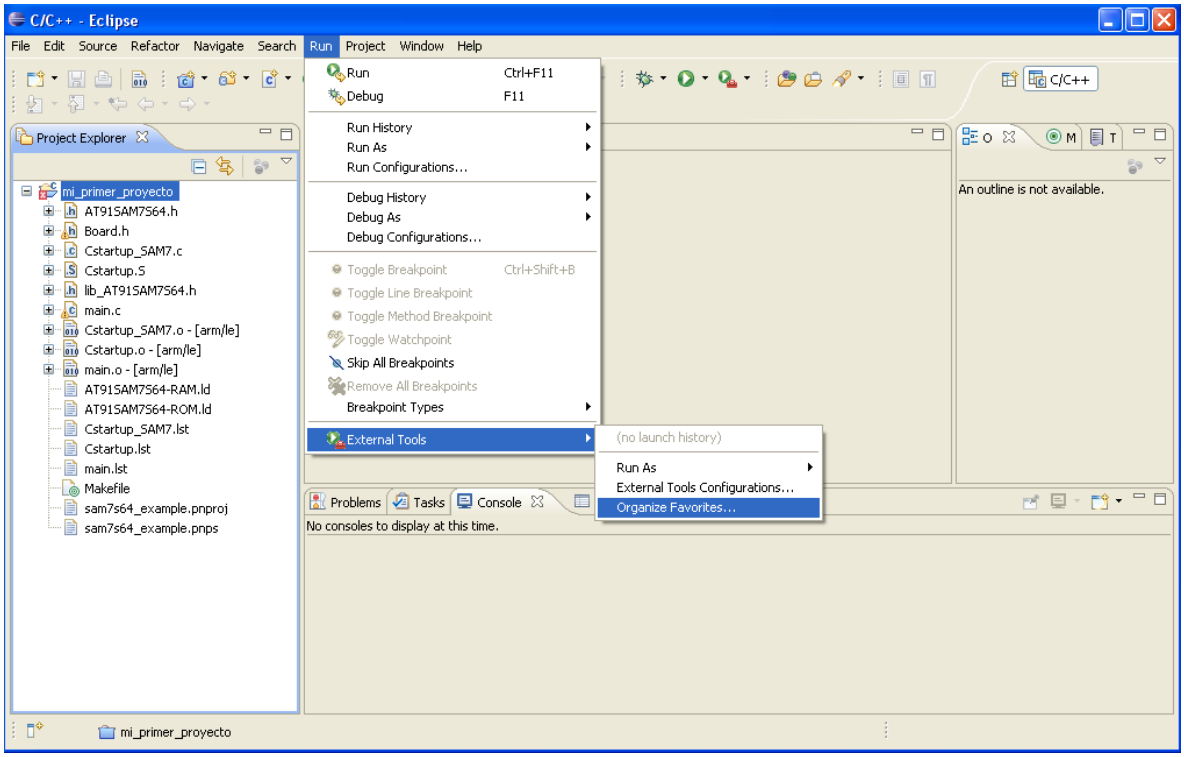


Los campos faltantes se pueden dejar vacios, y procedemos aplicar (*apply*) y luego a cerrar la ventana dando click en *close*.

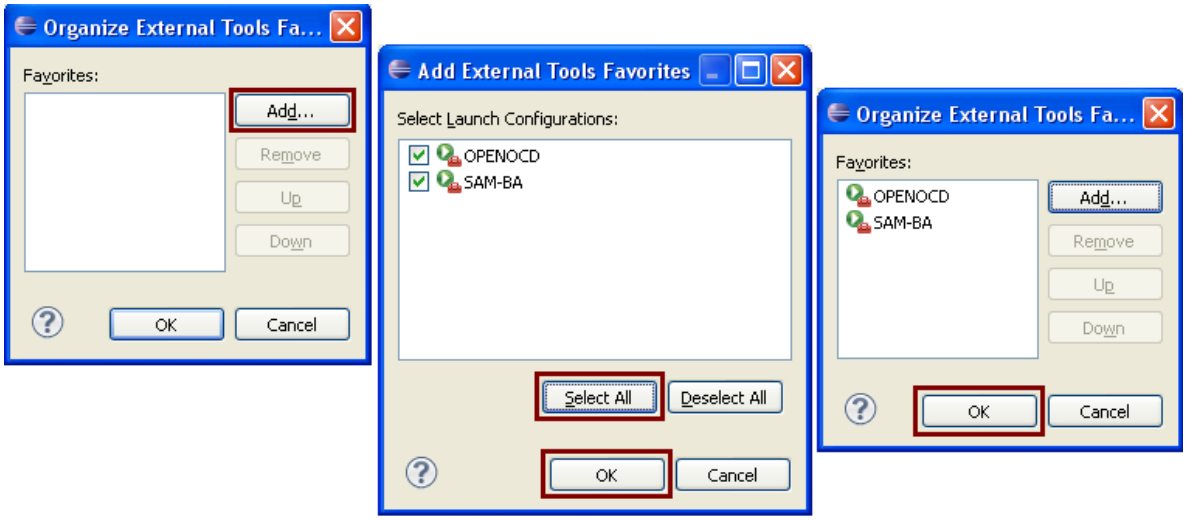


Organizando las herramientas externas favoritas: Para mayor facilidad y rapidez en el uso de las herramientas externas como SAM-BA u OpenOCD.

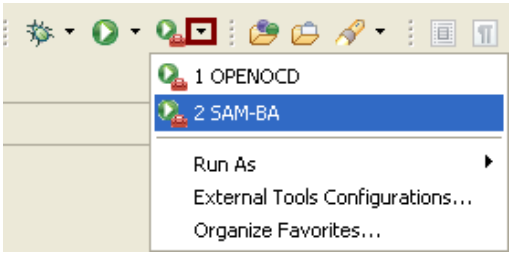
Ir al menú *Run -> External tools -> Organize favorites…*



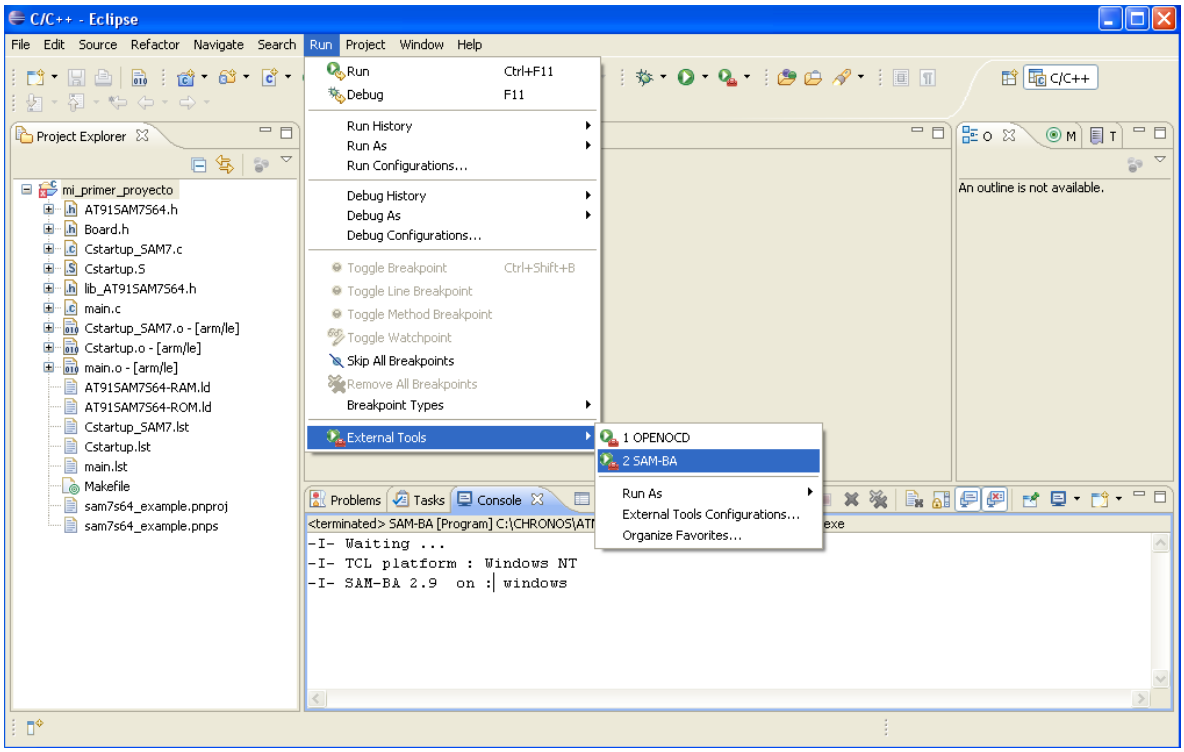
Luego, en la ventana que se abre adicionar las herramientas OpenOCD y SAM-BA y luego dar click en *ok*.



Realizado esto, ya podremos acceder al SAM-BA y al OpenOCD mas ágilmente tan solo con clickear en la flechita del botón *Run* (si se presiona el icono y no la flecha Eclipse correrá la herramienta que haya sido usada por última vez)



Otra forma de acceder a estas herramientas (en caso de no tener activada la barra de herramientas correspondiente) es ir al menú *Run -> External tools ->* y seleccionar la herramienta a utilizar (habitualmente será SAM-BA)

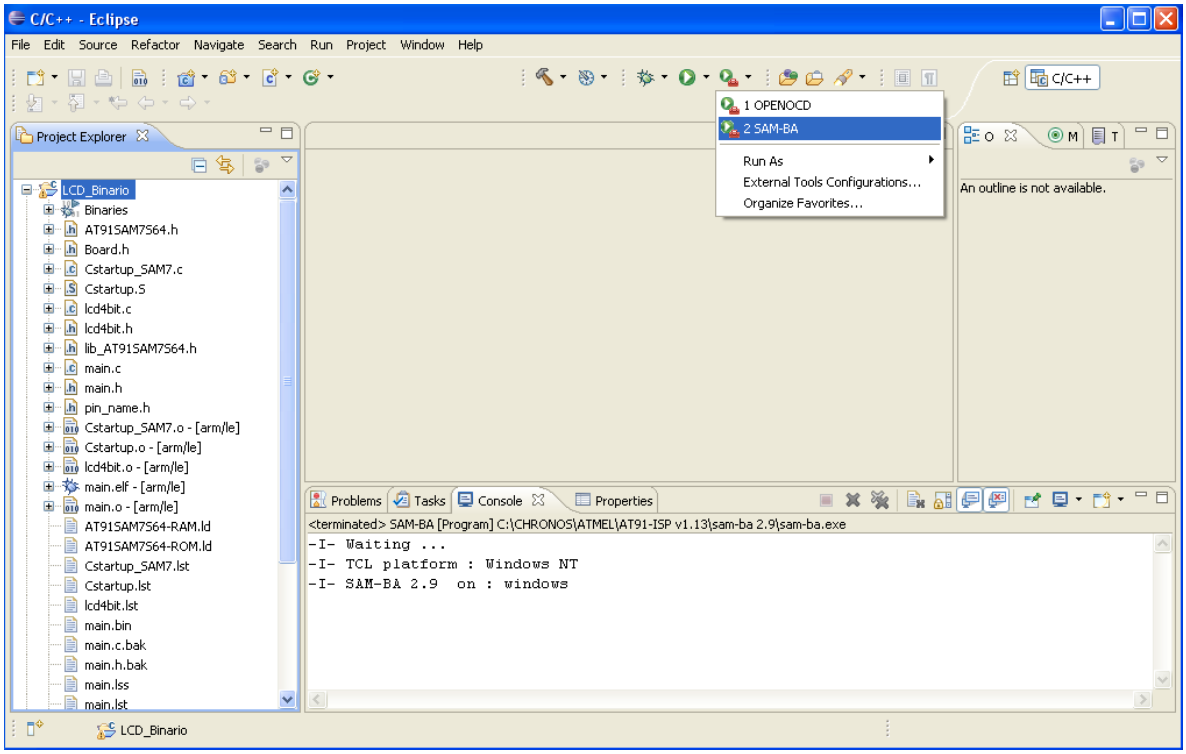


Programación:

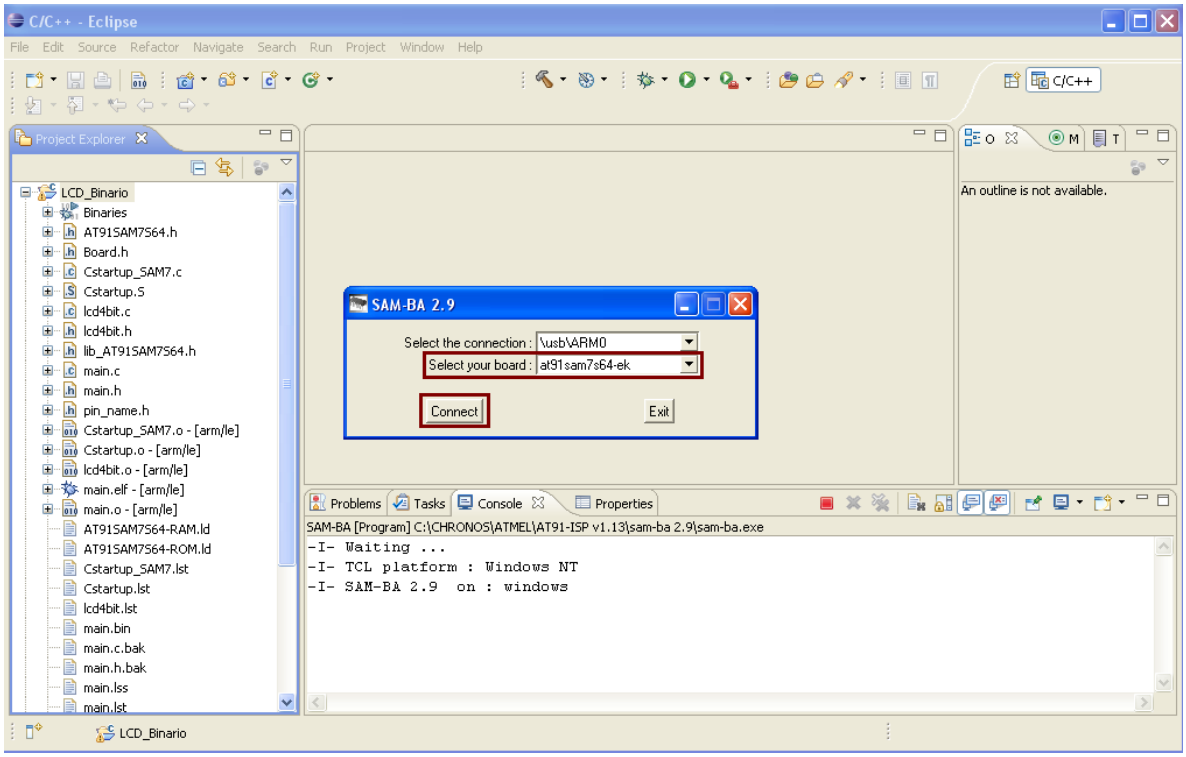
Luego de realizar correctamente los procesos anteriores, que solo necesitan ser efectuados una sola vez) describiremos el procesos de programación (que es necesario hacer cada vez que se vaya a programar el microcontrolador).

Luego de haber creado, limpiado y construido exitosamente el proyecto en Eclipse, encender y conectar la tarjeta base con el cable USB al PC, este la reconocerá inmediatamente (si no es así volver a revisar este manual desde el principio).

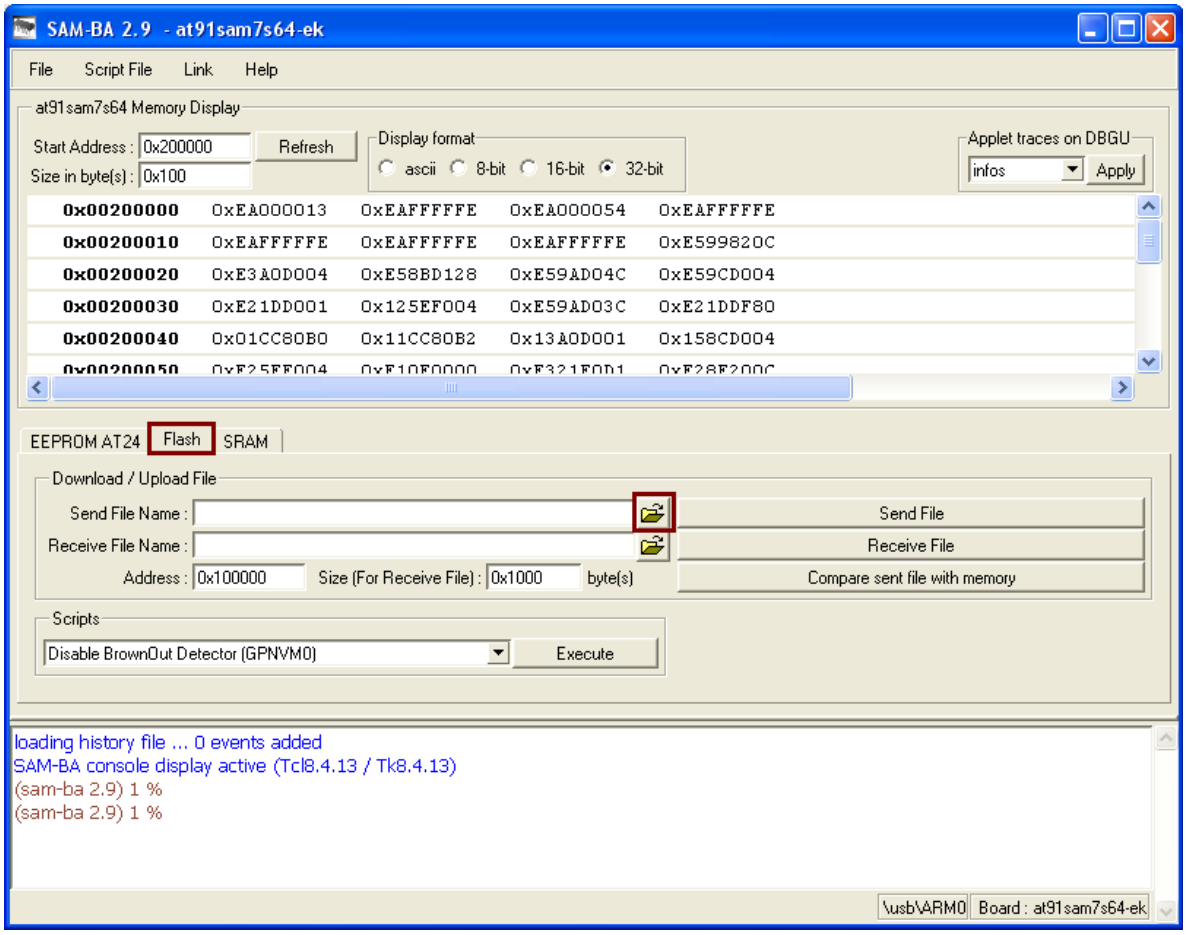
Vamos entonces a correr el SAM-BA como se describió anteriormente.



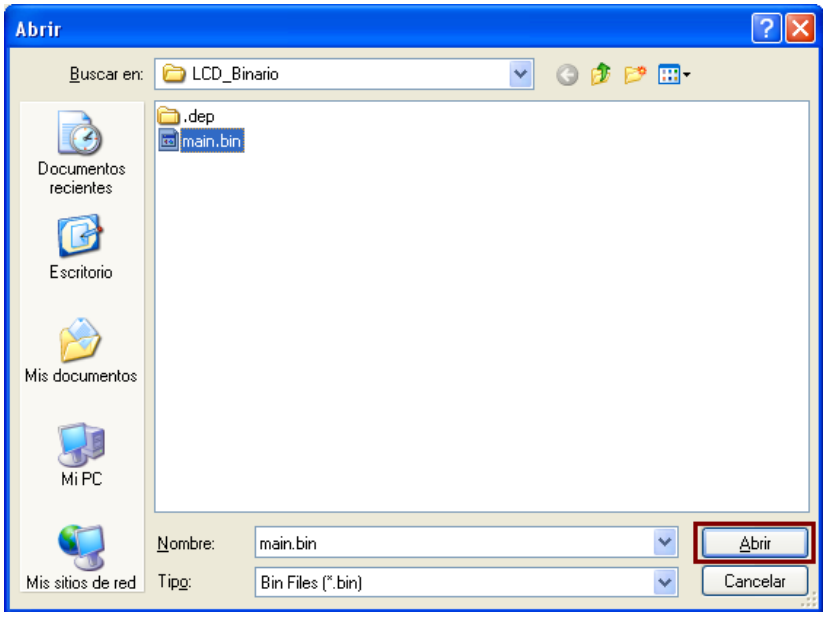
Eclipse tardará unos segundos mientras llama a la aplicación externa y aparecerá entonces la siguiente ventana, en donde se puede constatar que la tarjeta base fue reconocida correctamente pues aparece una conexión *usb/* de un dispositivo *ARM#* (el número representa cuantos hay conectados, y ARM es la arquitectura de nuestro microcontrolador). En el campo *select your board* es necesario elegir la referencia del microcontrolador que esté conectado a la tarjeta base, para este caso es una AT91SAM7S64, luego dar click en *connect.*



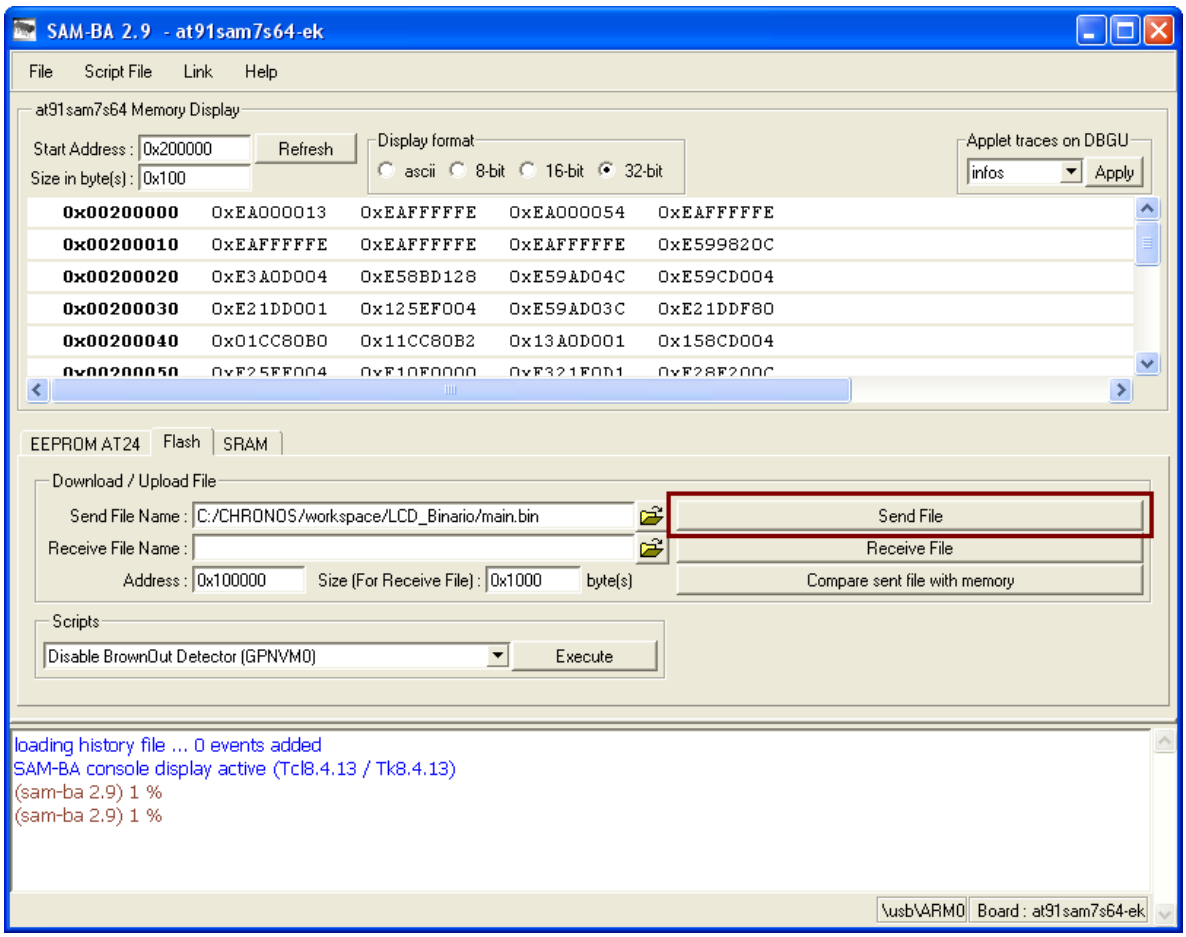
Aparecerá una ventana donde seleccionaremos la memoria del microcontrolador que se va a programar, que es la *Flash*, y tres campos principales, el primero y más grande que es donde se observa los datos que tiene el microcontrolador actualmente en su memoria, luego un campo con el que se le envían los archivos (binarios) al microcontrolador, y el último donde se recibe un archivo del microcontrolador que muestra todos los datos que tiene el microcontrolador en memoria. Así pues, como deseamos programar el microcontrolador, vamos a seleccionar el archivo que se le va enviar dando click en el icono de la carpeta al frente del botón *Send file*.



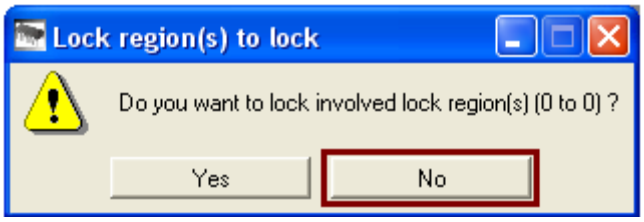
En la ventana que nos aparece buscar la carpeta del proyecto que hicimos en eclipse y que queremos programarle al microcontrolador(esta se encuentra dentro de la carpeta worspace, que Eclipse creó cuando se ejecutó por primera vez), y dentro de la carpeta del proyecto seleccionar el único archivo .bin, y que generalmente se va a llamar main.bin



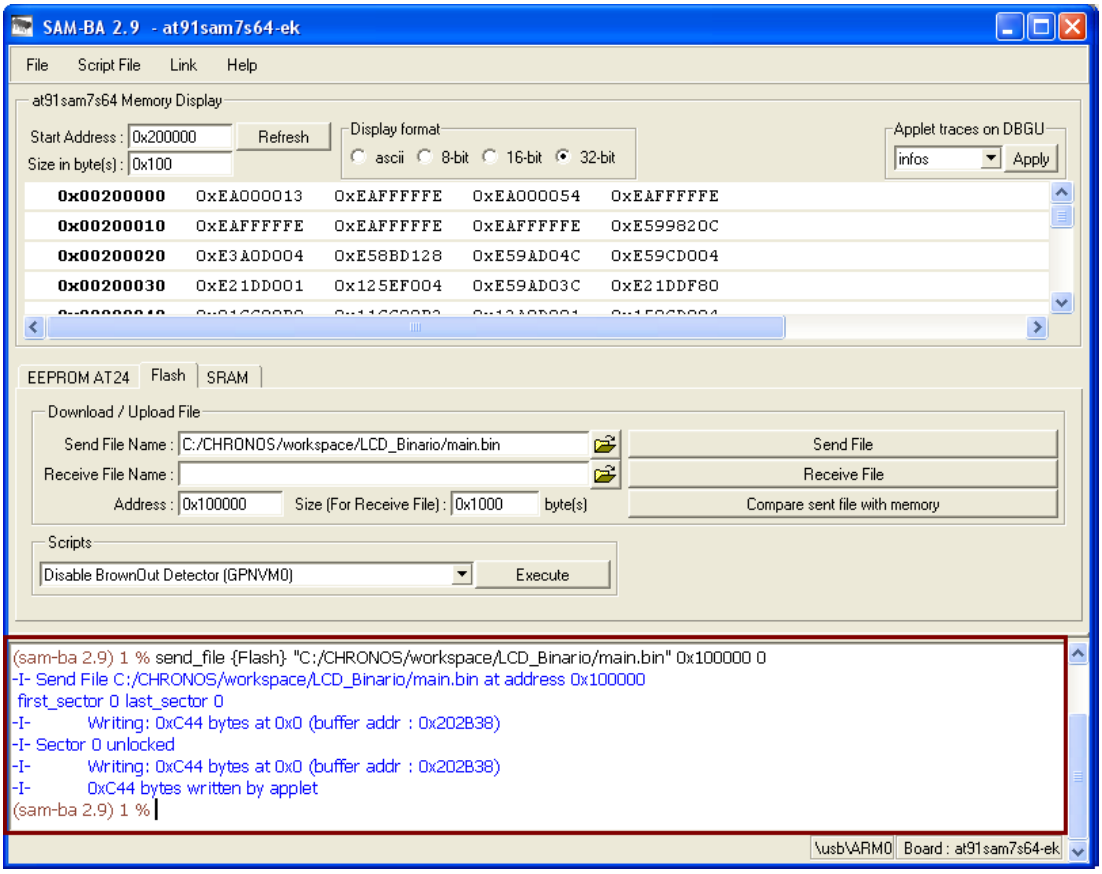
Luego ubicar el archivo binario que queremos programar, seleccionamos el botón *Send file.*



Nos aparecerá entonces un par de ventanas relacionadas con unas regiones bloqueadas, que son una protección que tiene el microcontrolador, a las cuales debemos responder *yes* (la que pregunta por si queremos desbloquear la región bloqueada), y a la siguiente *no* (la que pregunta si queremos bloquear de nuevo la región que acabamos de desbloquear), tal y como se aprecia en la siguiente grafica.



En la parte inferior del programa SAM-BA (la consola), aparecerá un mensaje diciendo que el microcontrolador se programó correctamente y cuantos bytes se programaron.



Una manera de comprobar si el microcontrolador quedó correctamente programado es pulsando en el botón *Compare sent file with memory*, si se programó bien la memoria del microcontrolador debe tener exactamente los mismos datos que el archivo binario que se envió, tal y como lo constata la ventana que aparece.