

Universidad Del Valle de Guatemala
Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica
Luis Alberto Rivera

Instalación de Eclipse CDT, GNU ARM Embedded Toolchain y TivaWare C para Windows 10 (64 bits)

Desarrollo de Programas para la Tiva C

Eclipse y MinGW

La instalación de Eclipse y MinGW se ilustra en el siguiente vídeo. Abajo se dan los enlaces para descargar los paquetes y se dan instrucciones adicionales.

<https://www.youtube.com/watch?v=WRYHKNV5H0>

Instalador de Eclipse (eclipse-inst-win64.exe):

<https://www.eclipse.org/downloads/packages/installer>

Nota: Correr el ejecutable puede tardar un poco.

Aparecerá una ventana para seleccionar el IDE específico que desea. Seleccione el “*Eclipse IDE for C/C++ Developers*”.

Instalar GCC (MinGW64): Esto permite compilar programas en C/C++ para Windows. Descargue el ejecutable [MinGW-W64-install.exe](https://sourceforge.net/projects/mingw-w64/files/mingw-w64/) (no el .zip) de

<https://sourceforge.net/projects/mingw-w64/files/mingw-w64/>

Al correr el instalador, en la ventana de *Settings*, posiblemente necesite cambiar la opción de “Architecture” a x86_64. Vea el vídeo indicado arriba (minuto 6:14). En la opción de versión, deje la versión más reciente.

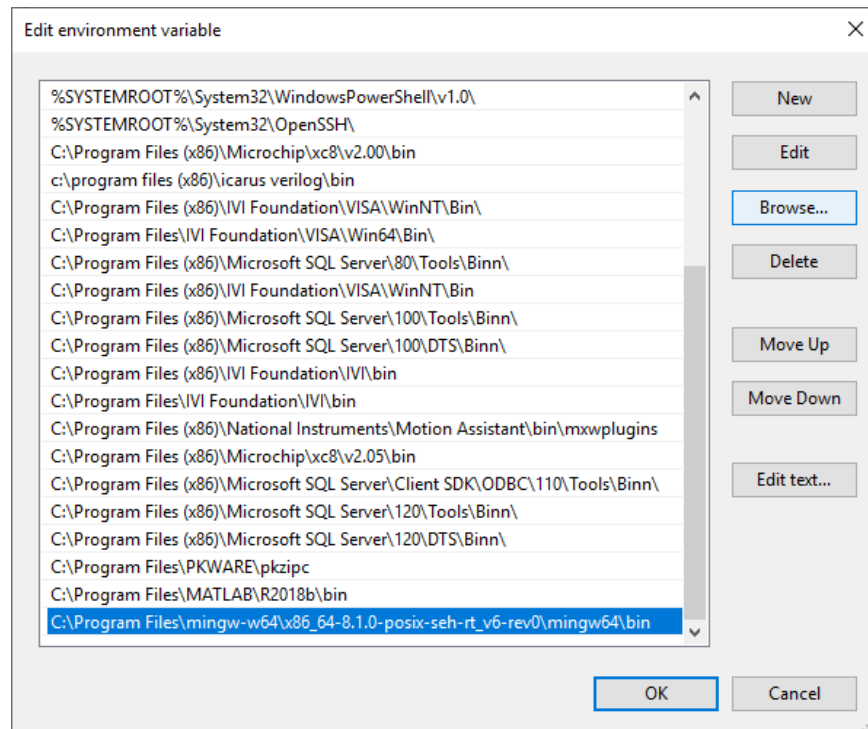
Agregar gcc/g++ de MinGW al “System Path”: Esto es para que Eclipse detecte a MinGW GCC automáticamente, y para poder compilar directamente desde una terminal fuera de Eclipse.

Siga las instrucciones de la página web siguiente, **a partir del paso #12:**

<http://www.codebind.com/cprogramming/install-mingw-windows-10-gcc/>

Nota: Probablemente la ruta que deba agregar sea distinta a la mostrada en la página web anterior. Allí se muestra “C:\MinGW\bin”, pero es probable que la ruta sea algo como “C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin”.

En la figura siguiente se muestra la ventana en la que debe agregar la ruta. Note que puede buscar el respectivo folder con el botón “Browse...”.



Para verificar que todo está bien, pruebe crear un programa “Hello World”, compilarlo y correrlo en Eclipse. Luego, pruebe correr el programa compilado en Eclipse en una terminal externa a Eclipse (*Command Prompt* y/o *Windows PowerShell*). Finalmente, pruebe compilar el programa directamente desde la terminal externa usando *gcc*. Corra el programa. Todo esto se ilustra en el documento “**Eclipse Hello World.pdf**”.

LM Flash Programmer, TivaWare C y GNU ARM Embedded Toolchain

Descargue e instale el LM Flash Programmer (LMFlashProgrammer.msi):

<http://www.ti.com/tool/LMFLASHPROGRAMMER>

Necesitará registrarse. El usuario creado de Texas Instruments le servirá para éste y otros productos.

Descargue e instale el TivaWare C para los controladores de la serie TM4C (SW-TM4C-2.1.4.178.exe):

http://software-dl.ti.com/tiva-c/SW-TM4C/latest/index_FDS.html

Puede ingresar con el mismo usuario creado anteriormente.

Instale el GNU ARM Embedded Toolchain (arm-none-eabi):

(descargue el .exe de la versión más reciente para Windows; no importa que diga 32 bits)

<https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads>

IMPORTANTE: Antes de terminar la instalación, marque la casilla “Add path to environment variable”, si ésta no está marcada por defecto.

Al parecer, en C:\Program Files (x86)\GNU Tools Arm Embedded\9 2019-q4-major\bin no está el ejecutable **make.exe**, necesario a la hora de compilar. Se puede usar el make.exe que trae MinGW64. Siga los siguientes pasos:

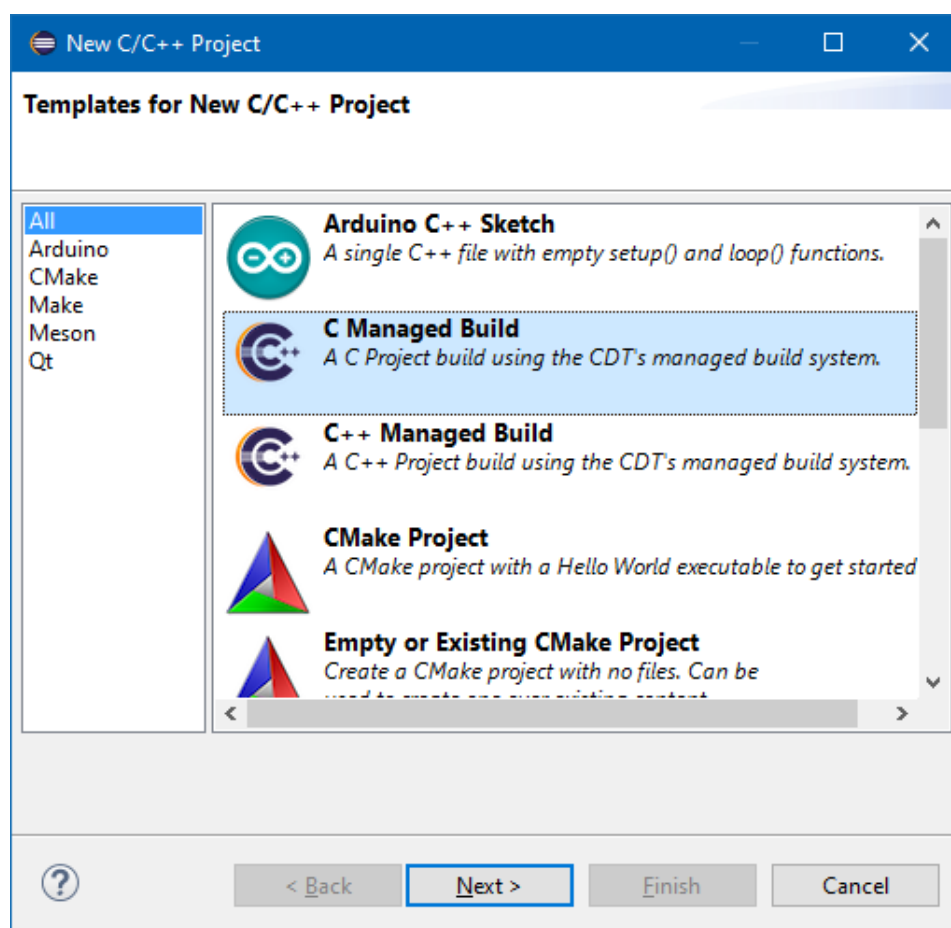
Vaya a: C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin
Debería ver el ejecutable “mingw32-make.exe”. Copie dicho archivo al directorio C:\Program Files (x86)\GNU Tools Arm Embedded\9 2019-q4-major\bin
Finalmente, cámbiele el nombre de “mingw32-make.exe” a “make.exe”.

Nota: las rutas indicadas arriba se asumen por defecto. Éstas pueden diferir, según su instalación de GNU ARM Embedded Toolchain y MinGW.

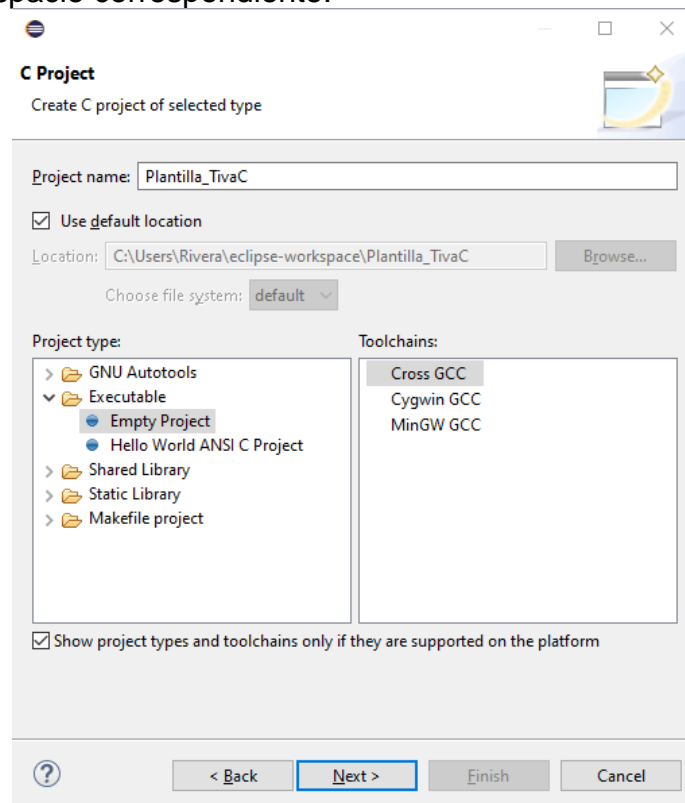
Desarrollo de Programas para la Tiva C en Eclipse:

Abra Eclipse y seleccione el *workspace* en que se desee trabajar.

Cree un proyecto nuevo: en el menú principal, *File > New > C/C++ Project*. En la ventana que aparece, seleccione la opción “C Managed Build”.



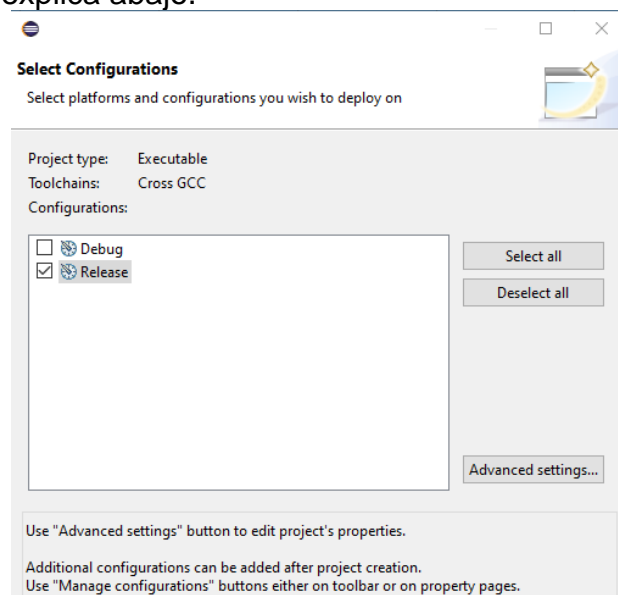
Seleccione el tipo de proyecto “Empty Project” y el *toolchain* “Cross GCC”. Dé un nombre al proyecto en el espacio correspondiente.



En la ventana “Select Configurations”, remueva el cheque en el cuadro “Debug”, dejando únicamente la opción de “Release”. Dele click en “Finish”.

Nota 1: se pueden dejar las dos opciones, pero se deberá tener mucho cuidado para las configuraciones que se describen más adelante. También se puede dejar únicamente la opción “Debug”, si se desea usar el *debugger* de Eclipse.

Nota 2: En “Advanced settings” se pueden hacer las configuraciones que se describen a continuación, de una vez. Dichas configuraciones se pueden hacer después de crear el proyecto, como se explica abajo.

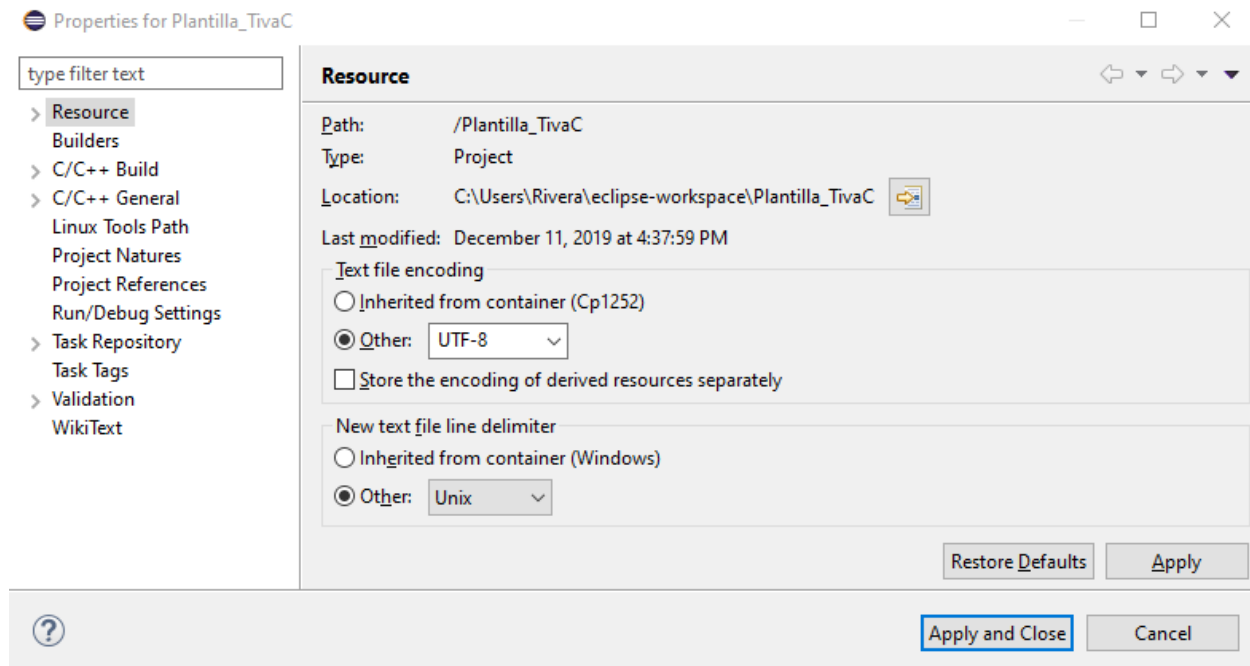


Deberá observar su proyecto creado en el “Project Explorer”. Dele click derecho en el proyecto, y busque la opción “Properties”. Alternativamente, puede presionar Alt+Enter, o puede ir al menú principal, *File > Properties*.

En la opción “Resource”:

Para “Text file encoding”, seleccione *Other > UTF-8*.

Para “New text file line delimiter”, seleccione *Other > Unix*



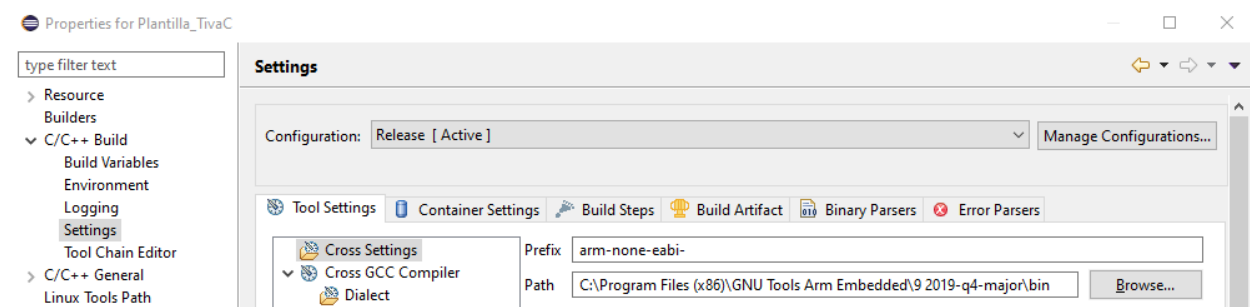
A continuación, expanda la opción “C/C++ Build”, y vaya a “Settings”.

En la opción “Cross Settings”:

Prefix: arm-none-eabi-

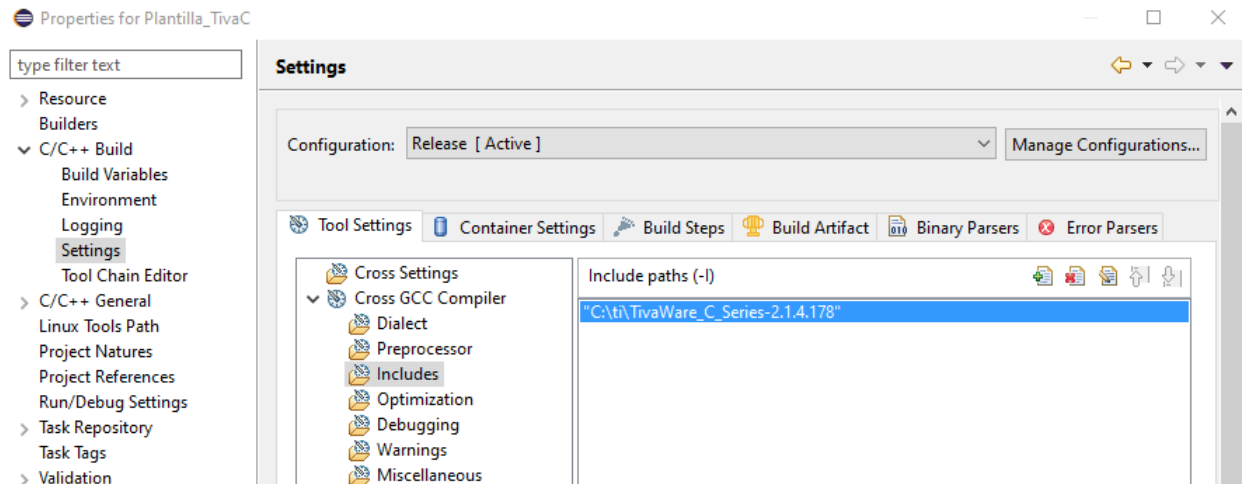
Path: C:\Program Files (x86)\GNU Tools Arm Embedded\9 2019-q4-major\bin

Nota: la ruta anterior puede cambiar, según lo especificado a la hora de instalar el GNU ARM Embedded Toolchain.

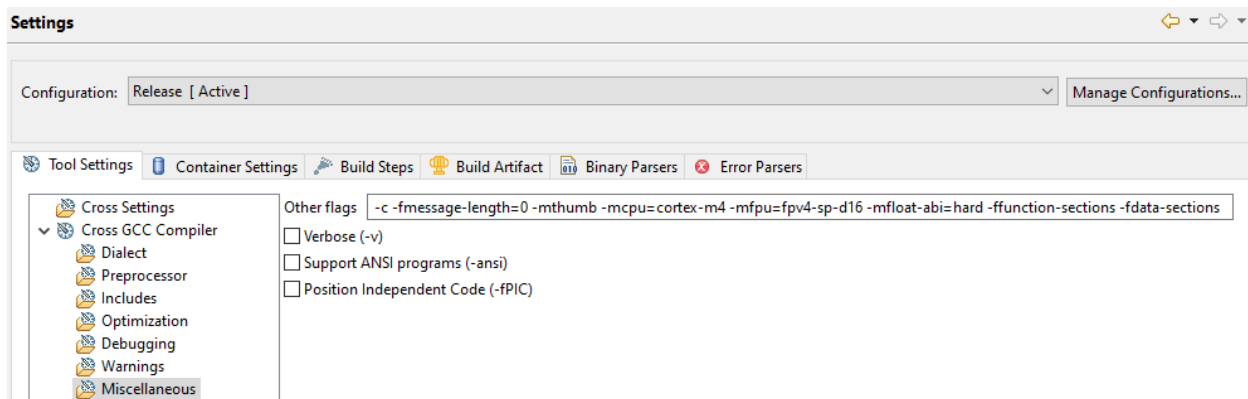


En *Cross GCC Compiler > Includes*,
incluya el *path* del TivaWare: C:\ti\TivaWare_C_Series-2.1.4.178

Nota: la ruta indicada es la ruta por defecto, pero depende de su instalación del TivaWare.



En *Cross GCC Compiler > Miscellaneous*, en “Other flags”, después de
-c -fmessage-length=0
agregue (después de un espacio en blanco)
-mthumb -mcpu=cortex-m4 -mfpv4-sp-d16 -mfloat-abi=hard -ffunction-sections -fdata-sections

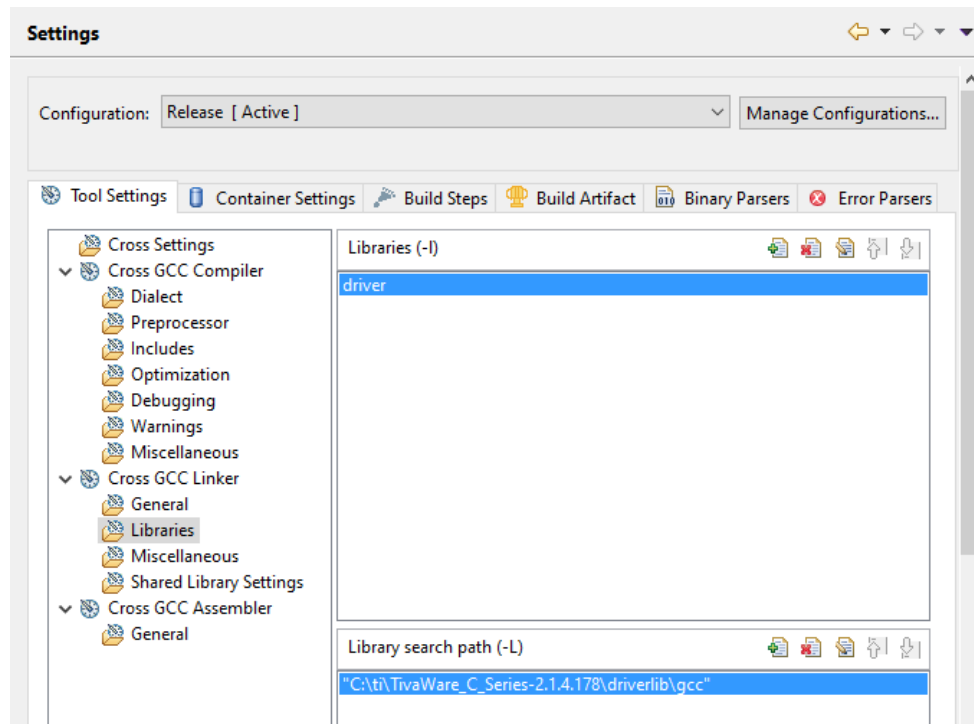


Necesitamos agregar el *TivaWare C driver library*.

En *Cross GCC Linker > Libraries*,

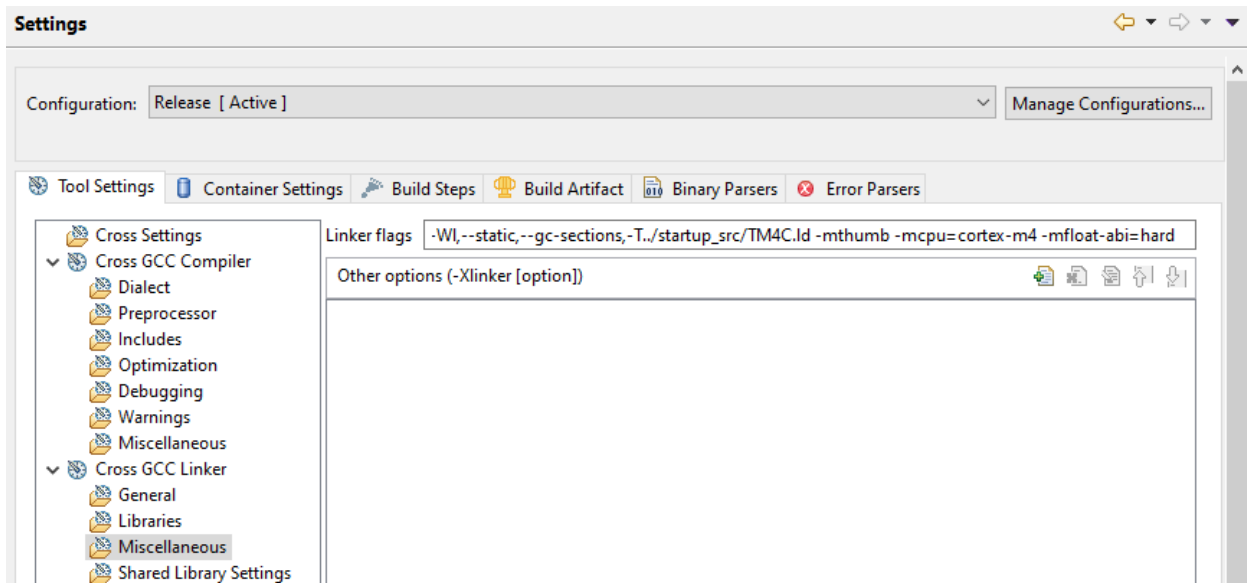
agregue `driver` bajo Libraries (-l),

y en Library Search path (-L), agregue `C:\ti\TivaWare_C_Series-2.1.4.178\driverlib\gcc`



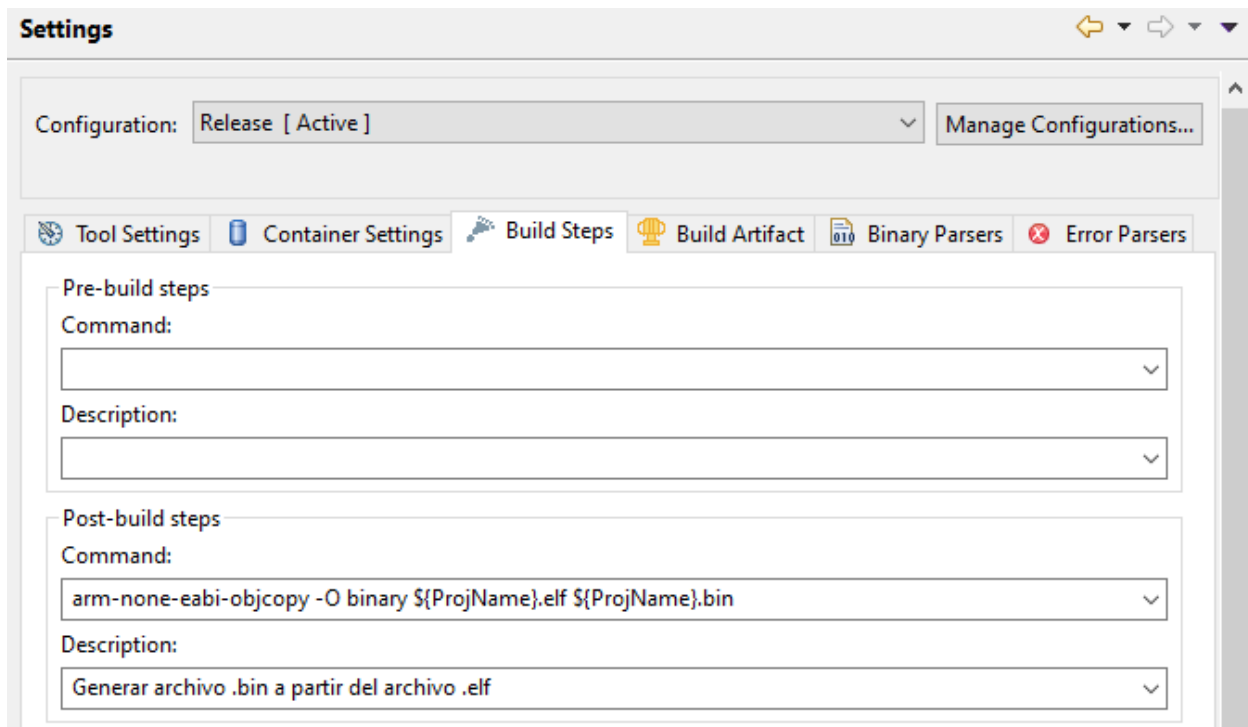
En *Cross GCC Linker > Miscellaneous*, en "Linker Flags", ingrese:

`-Wl,--static,--gc-sections,-T../startup_src/TM4C.ld -mthumb -mcpu=cortex-m4 -mfloat-abi=hard`

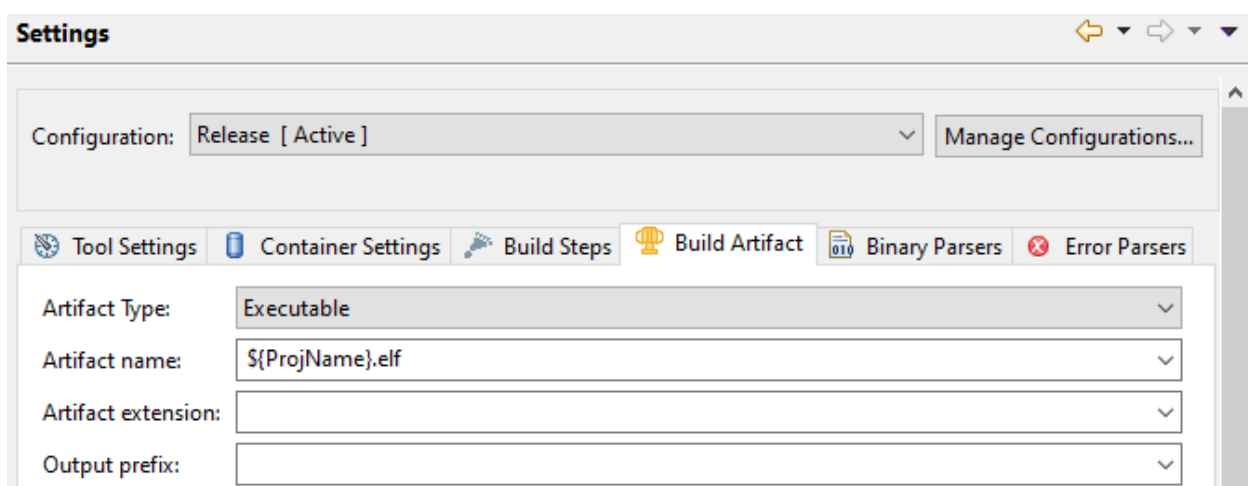


En la pestaña “Build Steps”,
 en *Post-build steps* > *Command*, ingrese:
`arm-none-eabi-objcopy -O binary ${ProjName}.elf ${ProjName}.bin`

En la descripción, se puede ingresar:
 Generar archivo .bin a partir del archivo .elf



En la pestaña “Build Artifact”, agregue .elf al “Artifact name” que aparece, \${ProjName}:

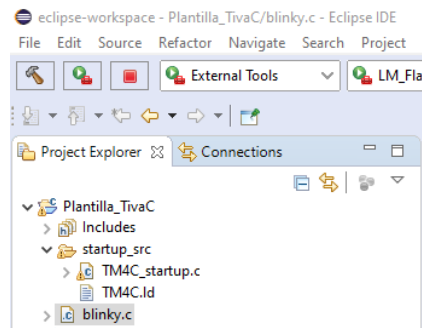


Dele click en “Apply and Close”.

Ahora necesitamos agregar archivos fuente al proyecto.

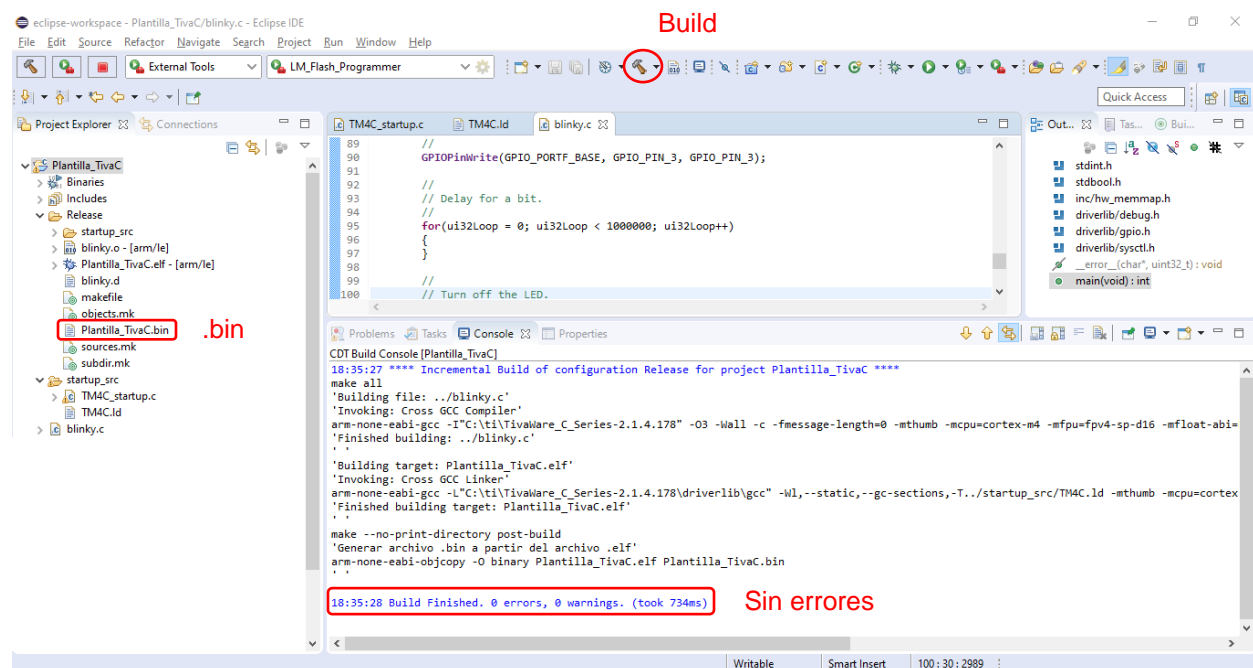
En primer lugar, descargue el folder “startup_src” que se encuentre en Canvas. Copie el folder “startup_src” al folder de su proyecto en el “Project Explorer” de Eclipse. “startup_src” tiene dos archivos, **TM4C_startup.c**, y **TM4C.ld**, los cuales tienen definiciones y declaraciones necesarias para inicializar su programa.

Finalmente, se debe agregar el archivo con su programa. Como ejemplo, copie el archivo **blinky.c** (también en Canvas). Este programa configura un LED de la Tiva C y lo enciende y apaga constantemente.



Con todo lo anterior, ya está listo para compilar su programa. Para compilar, busque el ícono de build (un martillo). Alternativamente, puede ir al menú principal y buscar **Project > Build Project**.

Si todo está bien, al compilar no aparecerán mensajes de error en la consola, y en el folder de su proyecto aparecerá el folder “Release” (y/o “Debug”, según la configuración de su proyecto). En “Release”, verá varios archivos, incluido un .bin. El archivo .bin es el que se cargará a la Tiva C, como se explica más adelante.



Para futuros programas, puede crear proyectos en blanco e importar archivos y configuraciones desde el proyecto anterior. Así, sólo deberá reemplazar/agregar los archivos .c. No necesitará hacer todas las configuraciones nuevamente.

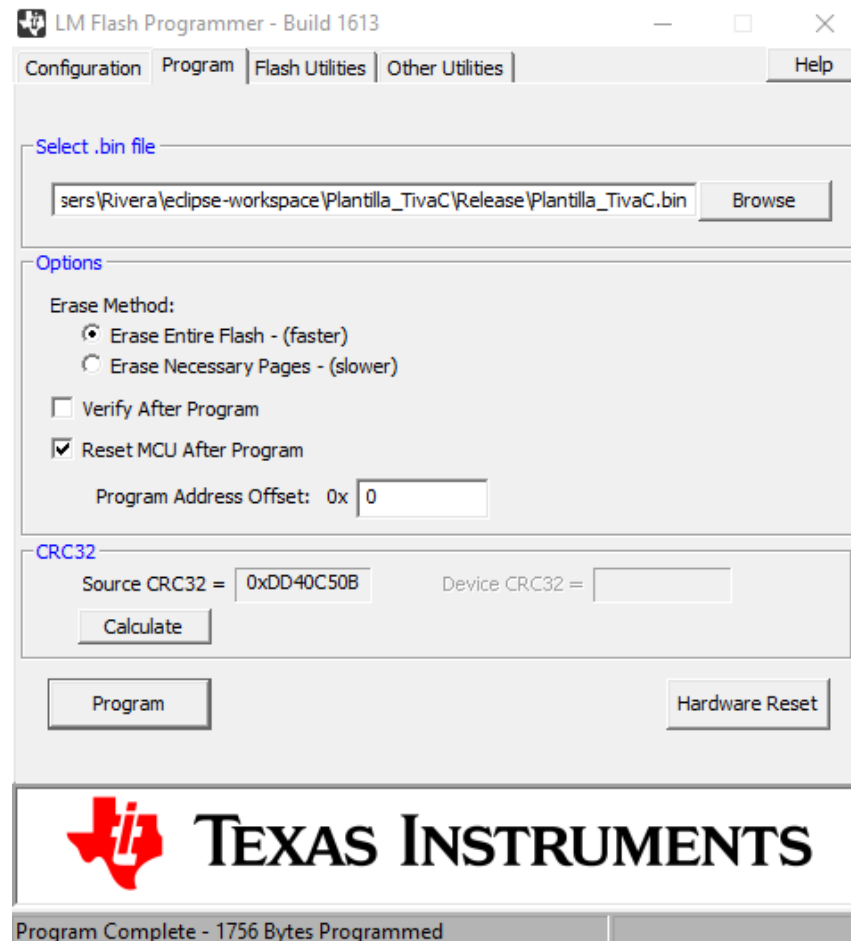
Para programar la Tiva C

Usando LM Flash Programmer directamente

Abra el programa.

En la pestaña “Program”, busque el archivo .bin generado en Eclipse. Se recomienda seleccionar la opción “Reset MCU After Program” (si no, deberá presionar el botón de Reset de la Tiva luego de programar, para que el programa empiece a ejecutar).

Dele click en el botón “Program”.



Usando LM Flash Programmer desde Eclipse

Vaya al menú de Eclipse en la parte superior. Busque *Run > External Tools > External Tools Configurations*.

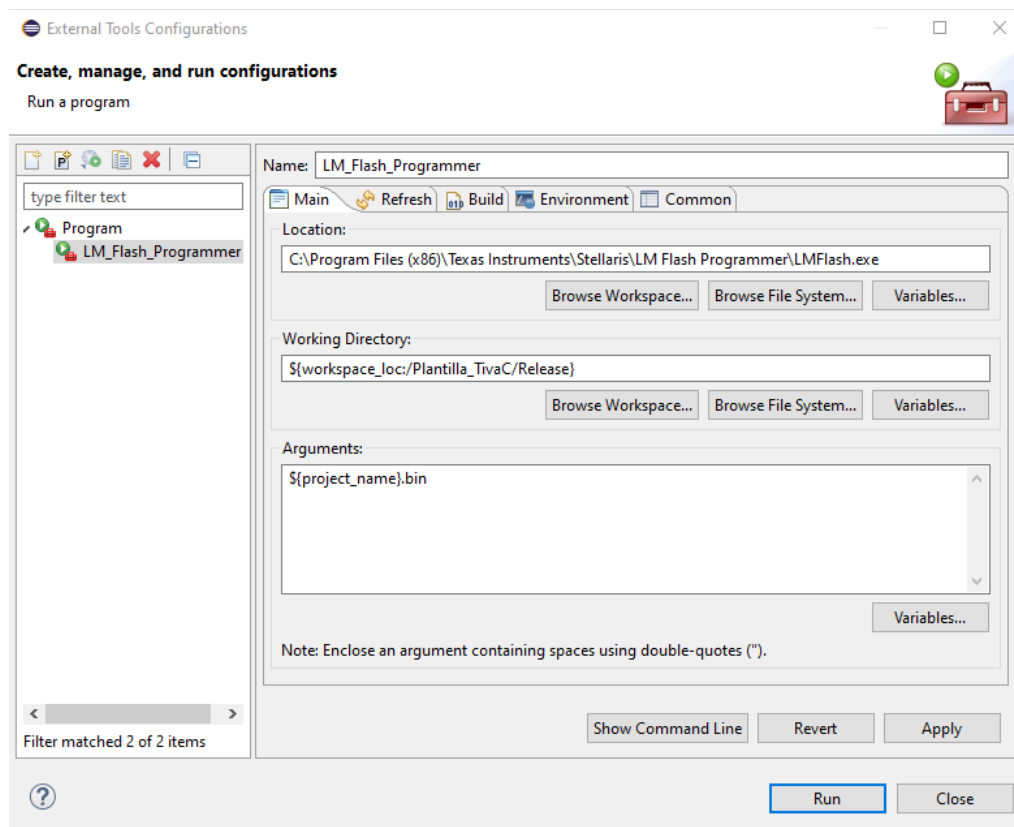
Dele click en “New launch configuration”

Nombre la configuración (se sugiere LM_Flash_Programmer)

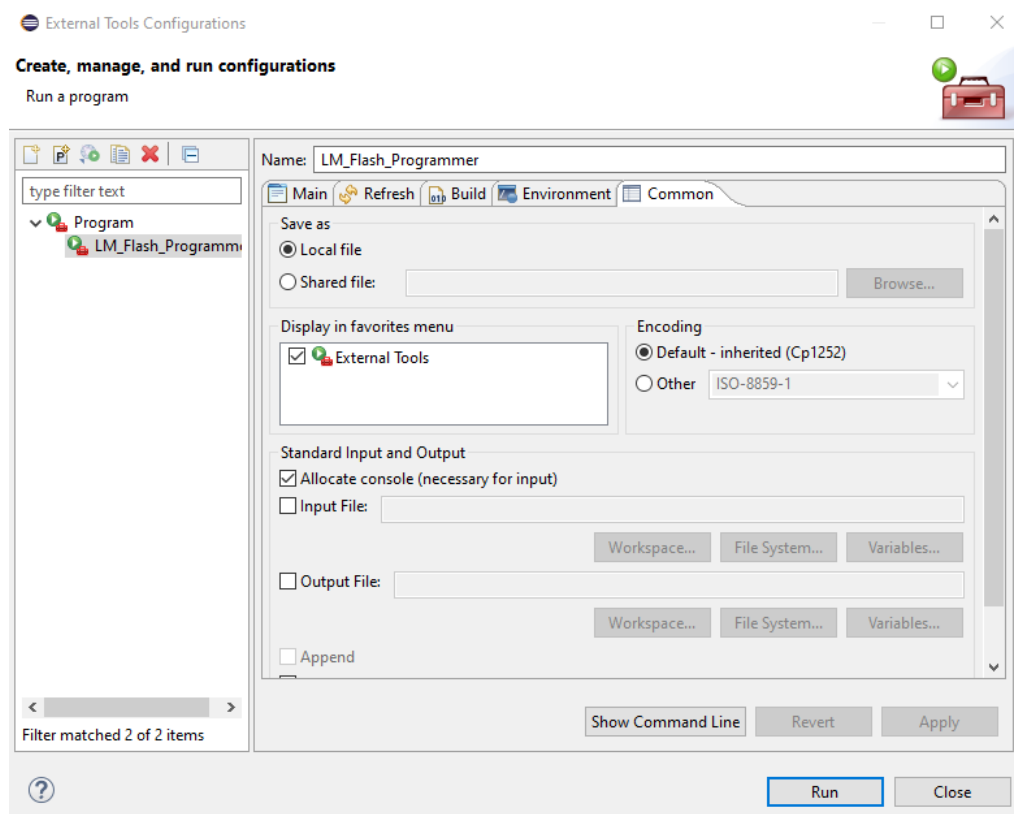
Busque la ubicación del LMFlash.exe y colóquela en “Location”.

En “Working Directory”, busque el directorio donde se encuentra el archivo .bin que se desea quemar en la Tiva.

En “Arguments”, coloque: \${project_name}.bin



Vaya a la pestaña “Common”.
En “Display in favorites menu”, marque la casilla de “External Tools”.



La próxima vez que se seleccione *Run > External Tools > LM_Flash_Programmer* o se presione el ícono de “Run LM_Flash_Programmer”, se compilará el programa y se cargará a la Tiva de una vez.

Importante: Luego de cargar el programa, deberá presionar el botón de *Reset* de la Tiva para que empiece a ejecutar.