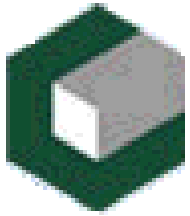
Software Development Kit
SDK

Maestría en Sistemas Digitales

Alejandro J. Cabrera Sarmiento

Dpto. de Automática y Computación
Universidad Tecnológica de La Habana
"*José Antonio Echeverría*"
CUJAE

alex@automatica.cujae.edu.cu

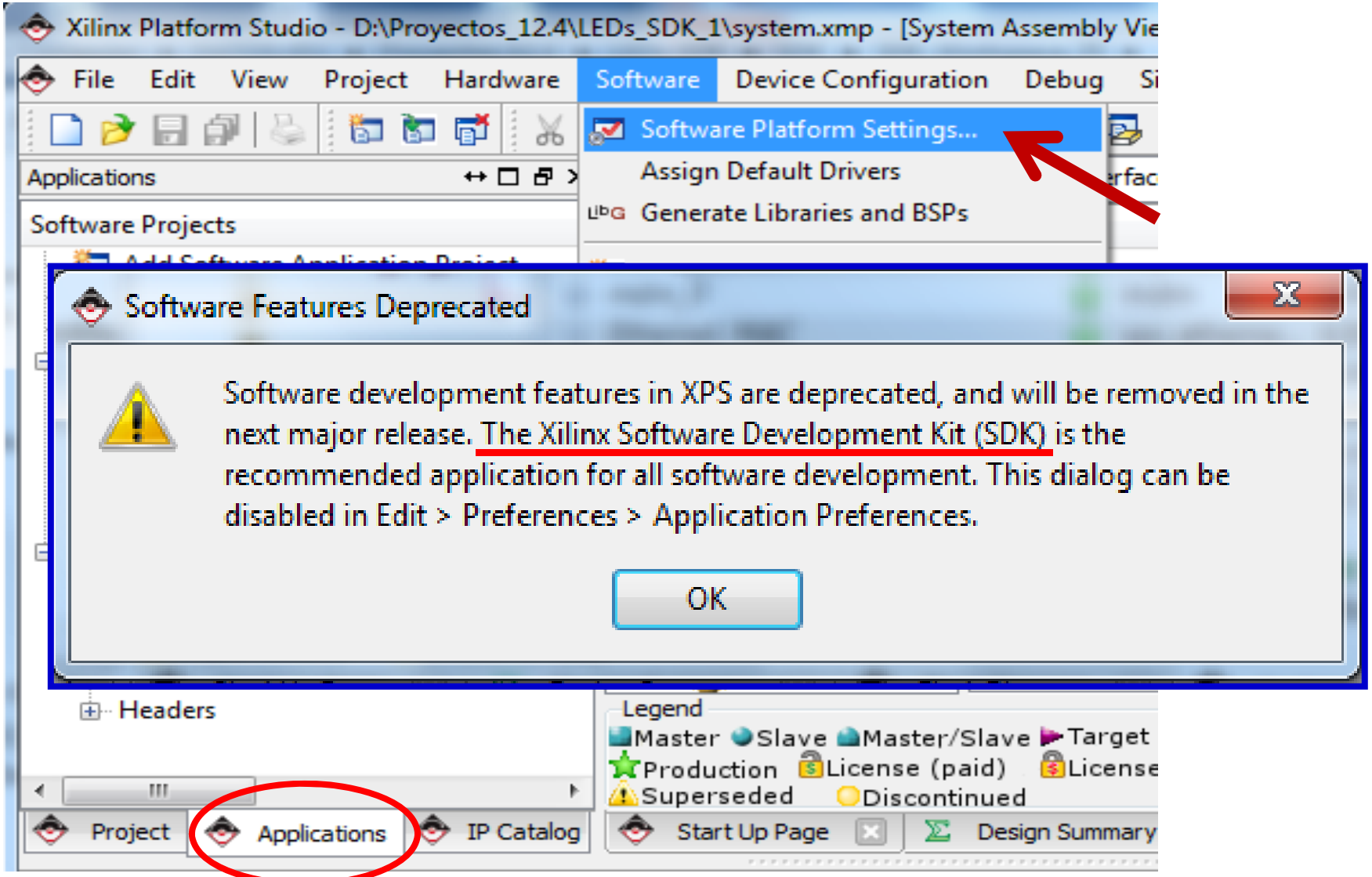


Sumario

- Características de SDK
- Desarrollo de aplicaciones con SDK
  - Exportar diseño HW
  - Configuración SW de la plataforma (BSP)
  - Creación de proyecto SW
  - Compilación y descarga al FPGA
- Depuración de aplicaciones con SDK

Desarrollo de aplicaciones

- **Puede hacerse en XPS hasta versión 12.4**



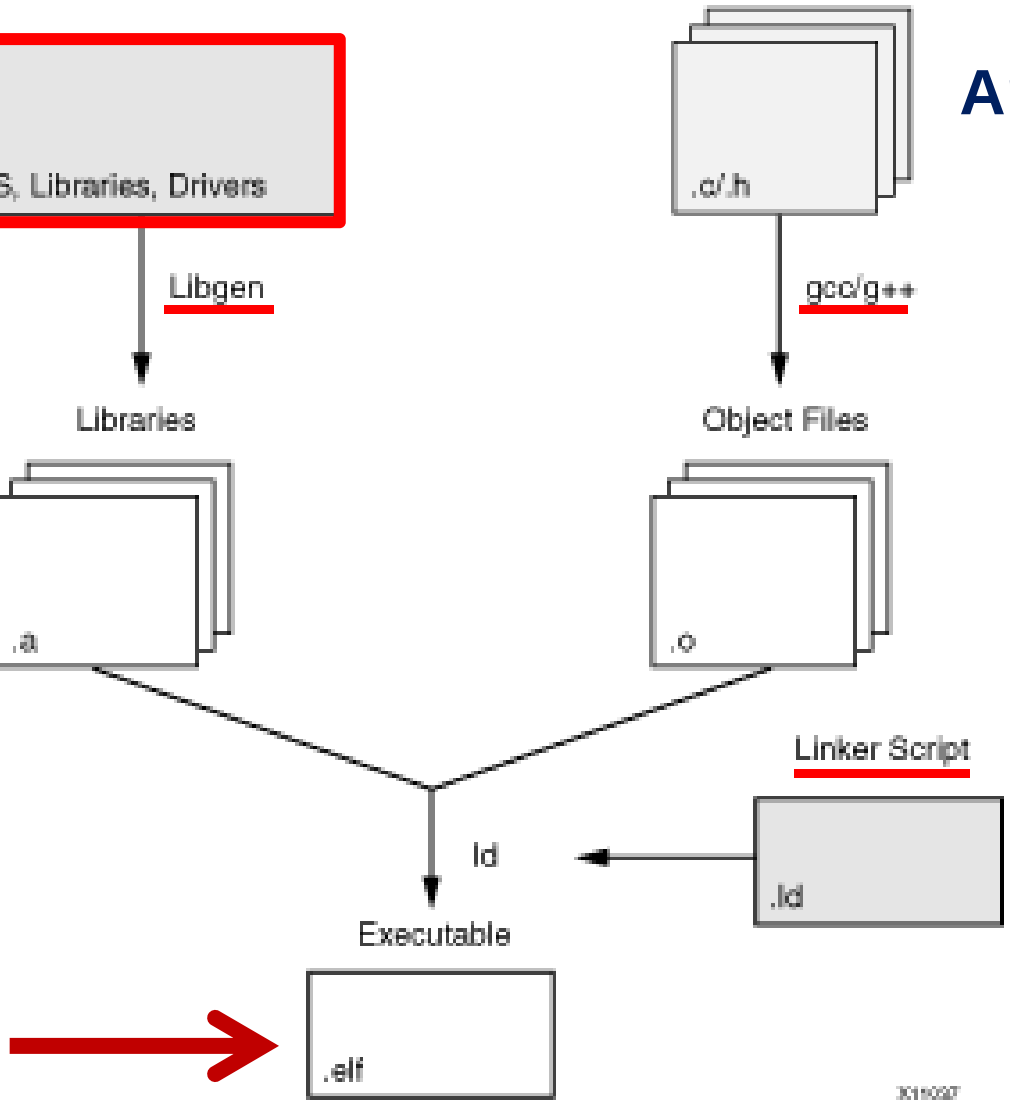
Software Development Kit

- **Entorno de desarrollo de aplicaciones SW**
- **Basado en Eclipse**
  - **Programa de código abierto multiplataforma para desarrollo de aplicaciones**
- **Facilidades:**
  - **Configurar la plataforma SW**
  - **Crear y compilar aplicación**
  - **Generar *download.bit***
  - **Descargar en el FPGA**
  - **Enlace con XMD**

Flujo de desarrollo SW con SDK

Plataforma SW

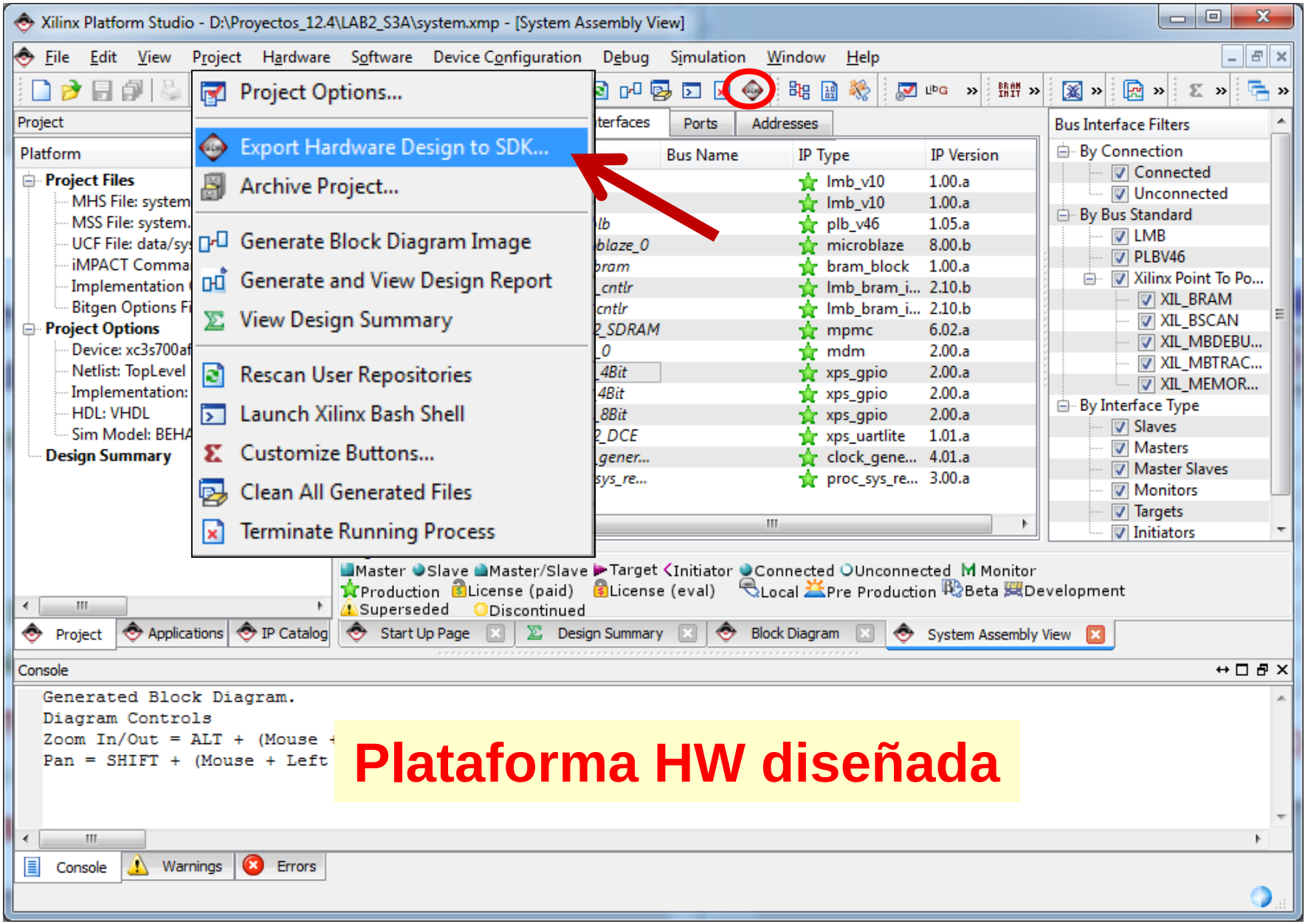
Aplicación SW



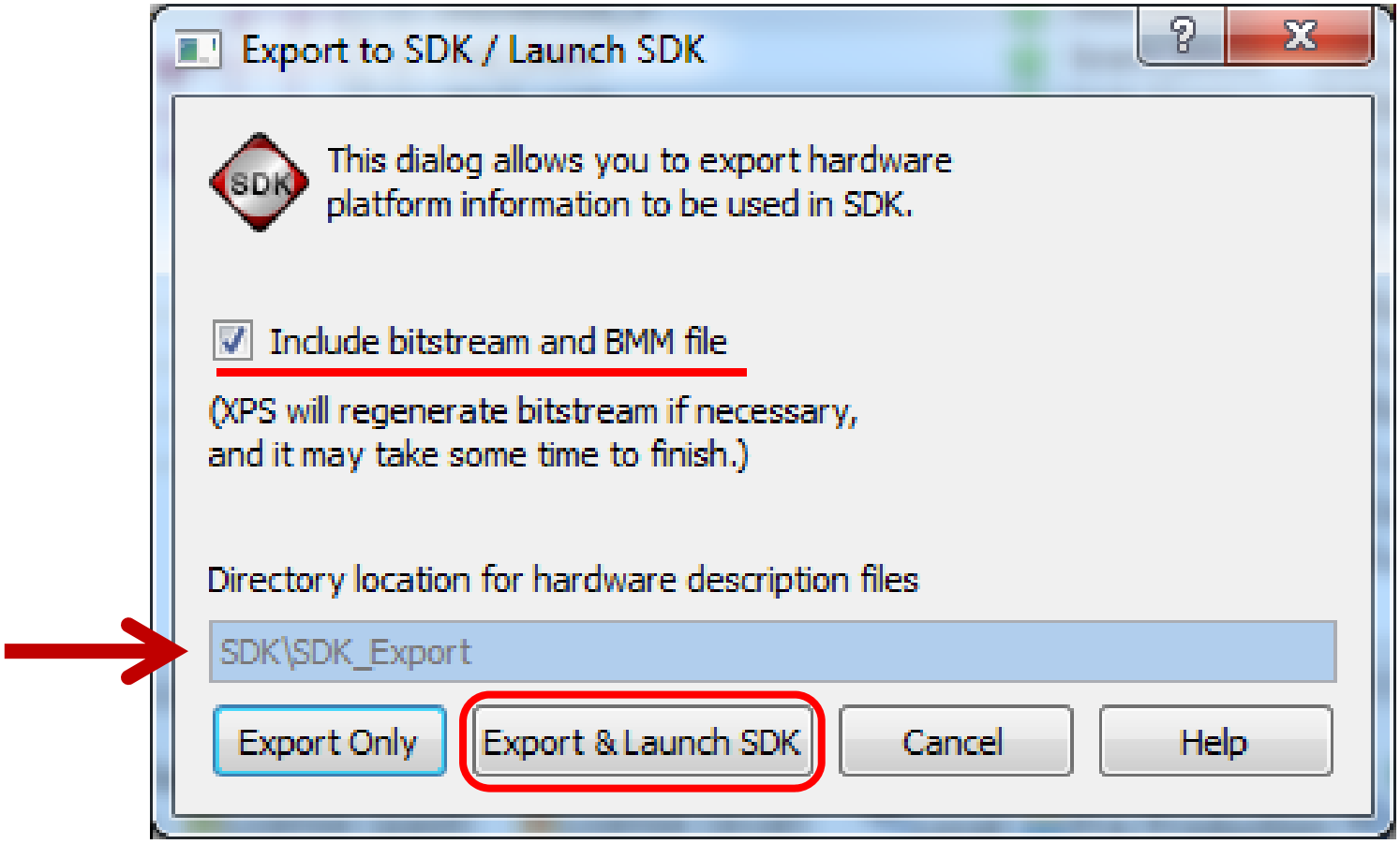
Flujo de diseño con SDK

1. Diseñar plataforma HW con XPS
2. Exportar diseño a SDK
   - Sintesis, implementación y generación de *system.bit*
   - Creación de espacio de trabajo *(Workspace)*
3. Configurar plataforma SW en SDK
   - Board Support Package: equivalente a *MSS*
   - Ejecuta *LibGen:* compilación de drivers, bibliotecas y SO
4. Crear proyecto con la aplicación SW de usuario
   - Contiene los ficheros fuentes de la aplicación
5. [Modificar el mapeo de memoria con linker script]
6. Implementar la aplicación SW
   - Compilación y enlace
7. Descargar el bitstream sobre el FPGA
   - Genera *download.bit*
8. Ejecutar y depurar el SW sobre el FPGA

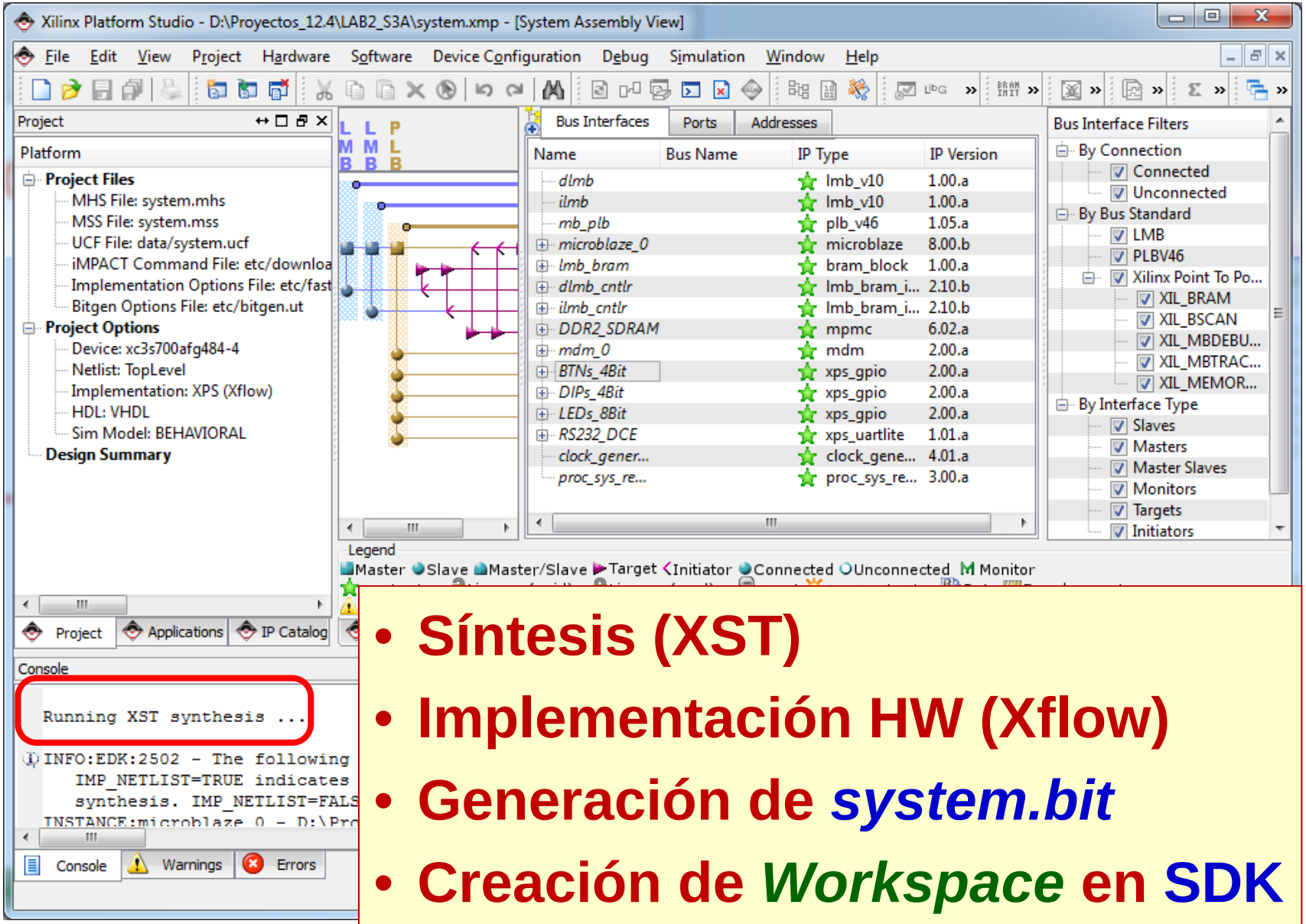
1. Diseño plataforma HW



2. Exportar HW a SDK

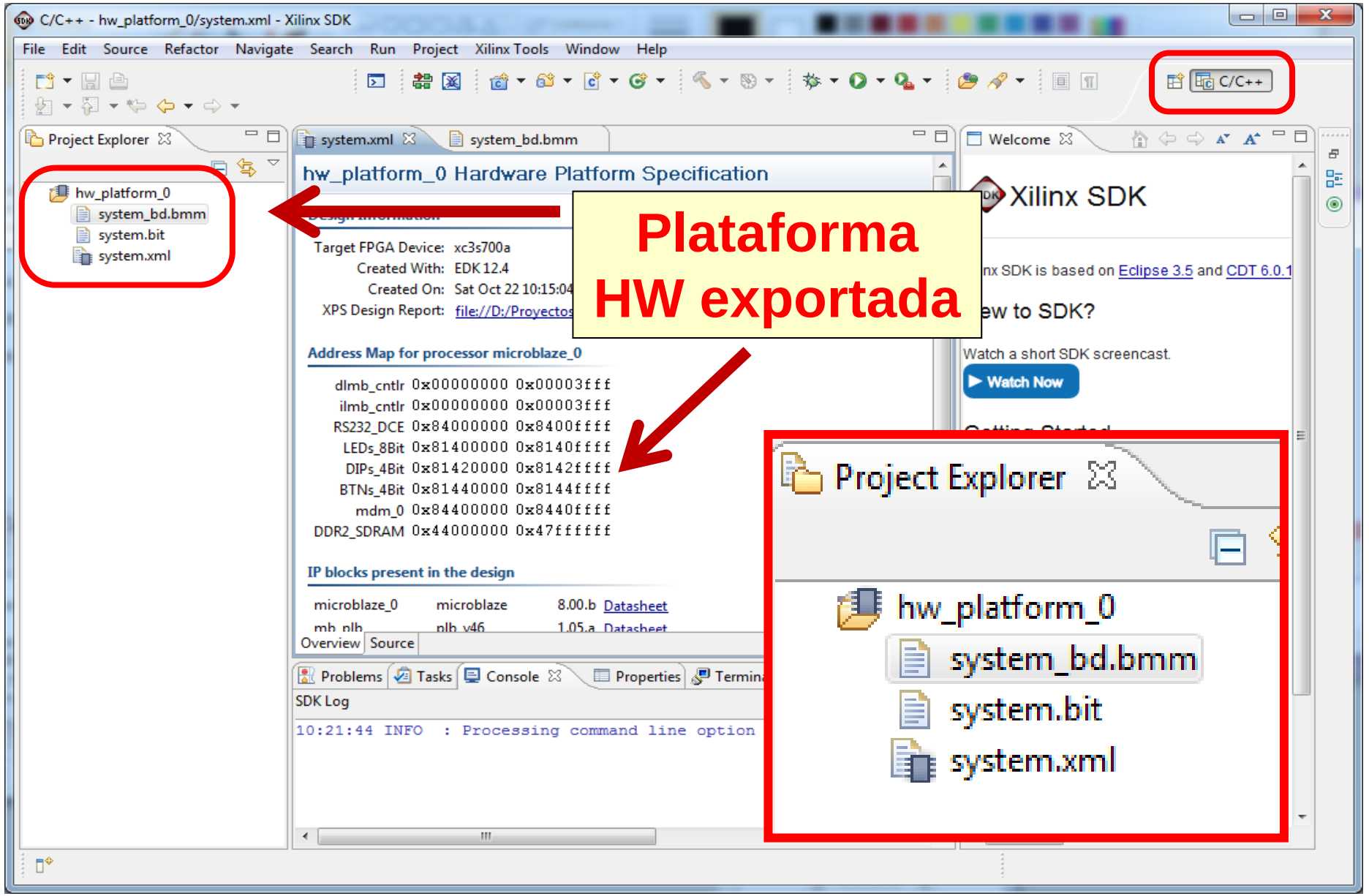


2. Exportar HW a SDK *(cont.)*



- **Síntesis (XST)**
- **Implementación HW (Xflow)**
- **Generación de *system.bit***
- **Creación de *Workspace* en SDK**

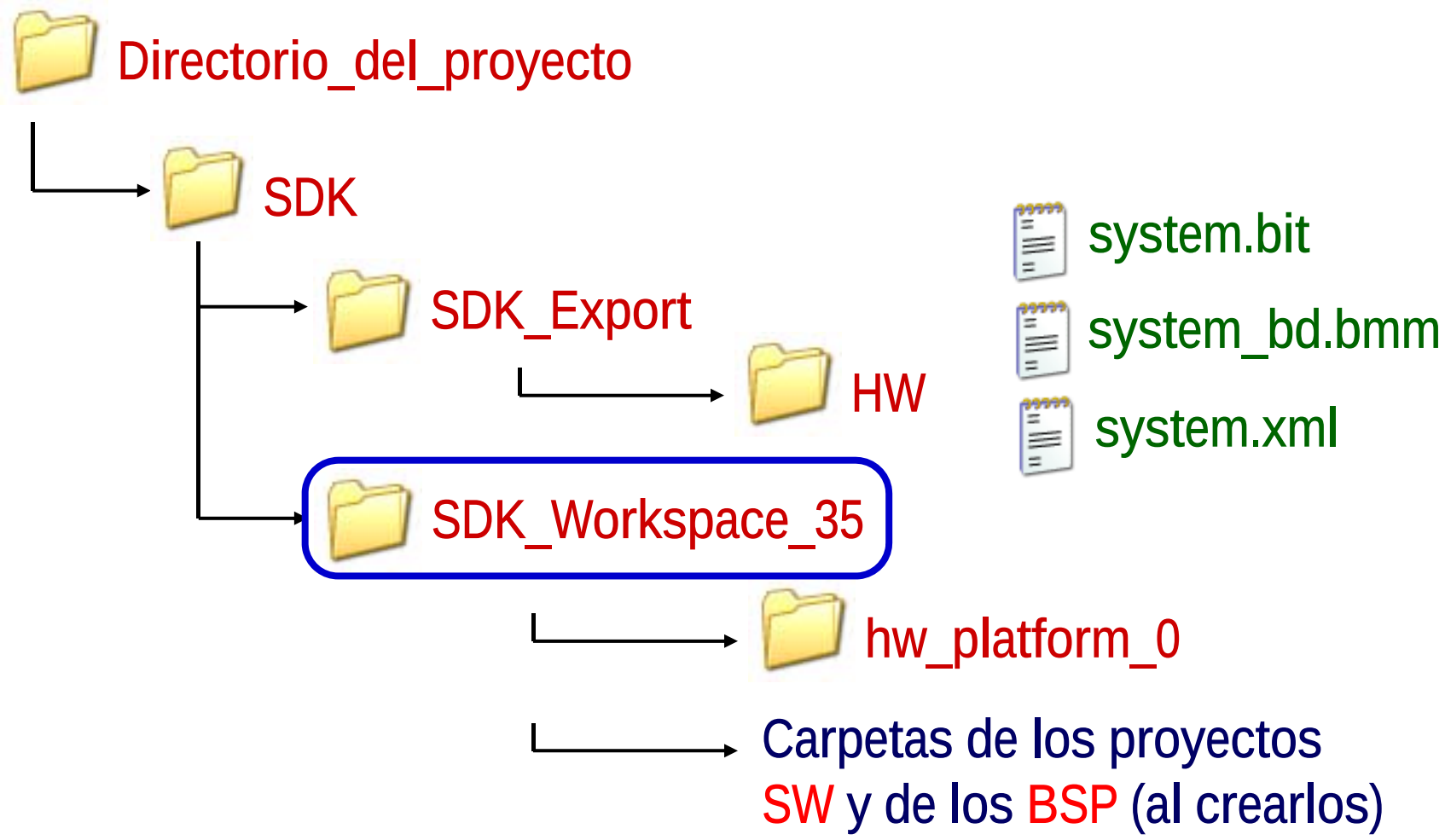
Interfaz de SDK



Workspace

- Directorio donde se almacena la información de un proyecto SDK
  - Se crea al abrir SDK
- Incluye:
  - Plataformas HW
  - Plataformas SW (Board Support Package)
  - Proyectos SW
- Pueden crearse varios Workspace

Estructura del proyecto



Plataforma HW

- **system-bd.bmm:** descripción de las conexiones de las BRAM utilizadas en la configuración HW.
  - Se utiliza para modificar el ***system.bit*** y obtener el ***download.bit*** en correspondencia con el programa.
- **system.bit:** interconexiones HW del sistema de procesamiento.
- **system.xml:** descripción HW del sistema de procesamiento resumido en un archivo XML
  - Equivalente a ***.mhs***

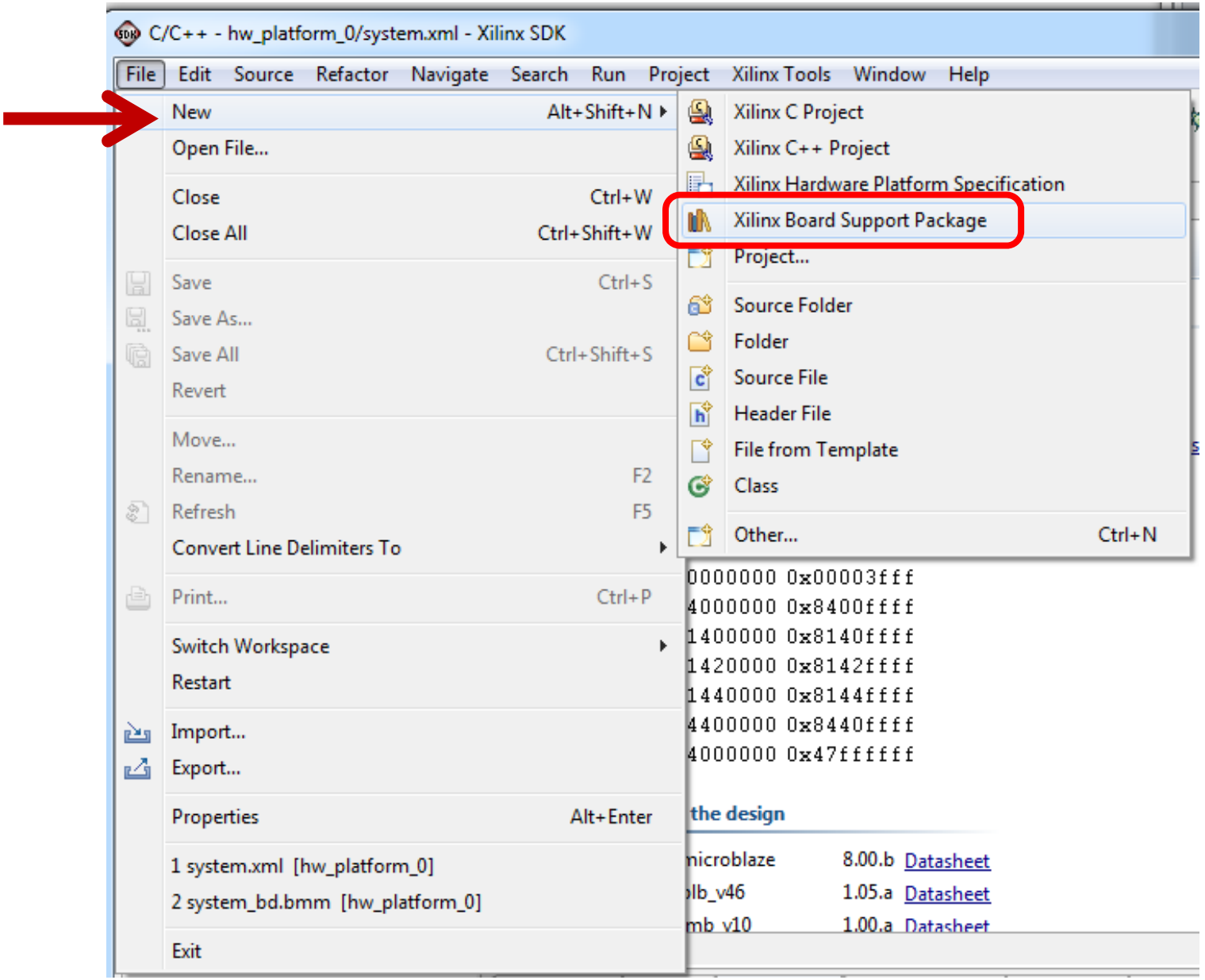
Fichero *system_bd.bmm*

system.xml | system_bd.bmm

```
// BMM LOC annotation file.
//
// Release 12.4 - Data2MEM M.81d, build 1.9 Aug 19, 2010
// Copyright (c) 1995-2016 Xilinx, Inc.  All rights reserved.


////////////////////////////////////////////////////////////////////////////////
//
// Processor 'microblaze_0', ID 100, memory map.
//
////////////////////////////////////////////////////////////////////////////////

ADDRESS_MAP microblaze_0 MICROBLAZE 100


    ////////////////////////////////////////////////////////////////////////////
    //
    // Processor 'microblaze_0' address space 'lmb_bram_combined' 0x00000000:0x00003FFF (16 KBytes).
    //
    ////////////////////////////////////////////////////////////////////////////

    ADDRESS_SPACE lmb_bram_combined RAMB16 [0x00000000:0x00003FFF]
        BUS_BLOCK
            lmb_bram/lmb_bram/ramb16bwe_0 [31:28] INPUT = lmb_bram_combined_0.mem PLACED = X1Y3;
            lmb_bram/lmb_bram/ramb16bwe_1 [27:24] INPUT = lmb_bram_combined_1.mem PLACED = X1Y2;
            lmb_bram/lmb_bram/ramb16bwe_2 [23:20] INPUT = lmb_bram_combined_2.mem PLACED = X0Y2;
            lmb_bram/lmb_bram/ramb16bwe_3 [19:16] INPUT = lmb_bram_combined_3.mem PLACED = X0Y3;
            lmb_bram/lmb_bram/ramb16bwe_4 [15:12] INPUT = lmb_bram_combined_4.mem PLACED = X1Y5;
            lmb_bram/lmb_bram/ramb16bwe_5 [11:8] INPUT = lmb_bram_combined_5.mem PLACED = X1Y7;
            lmb_bram/lmb_bram/ramb16bwe_6 [7:4] INPUT = lmb_bram_combined_6.mem PLACED = X1Y9;
            lmb_bram/lmb_bram/ramb16bwe_7 [3:0] INPUT = lmb_bram_combined_7.mem PLACED = X1Y8;
        END_BUS_BLOCK;
    END_ADDRESS_SPACE;
```

16 kB con
BRAM de 4k x 4

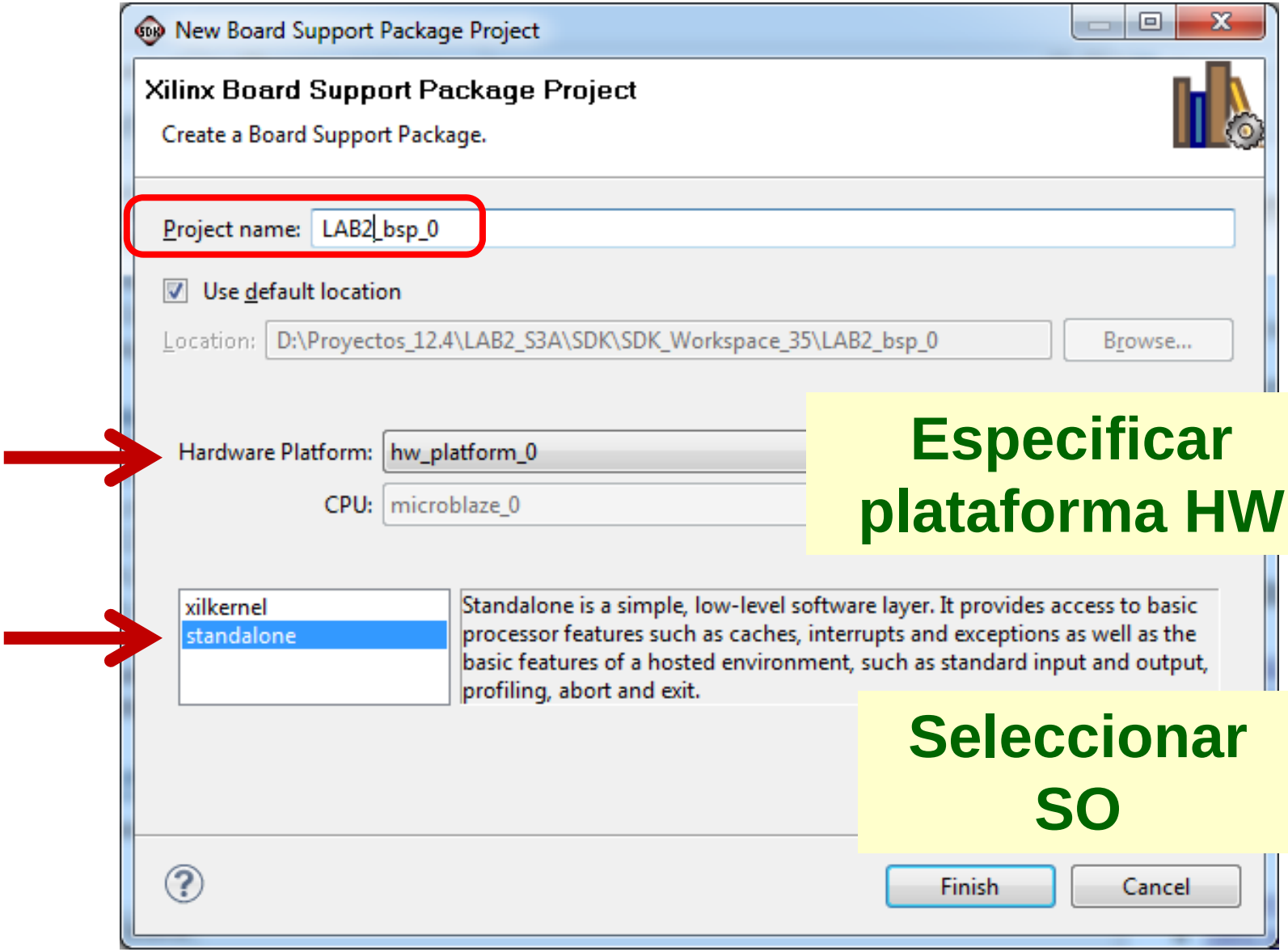
3. Configurar plataforma SW



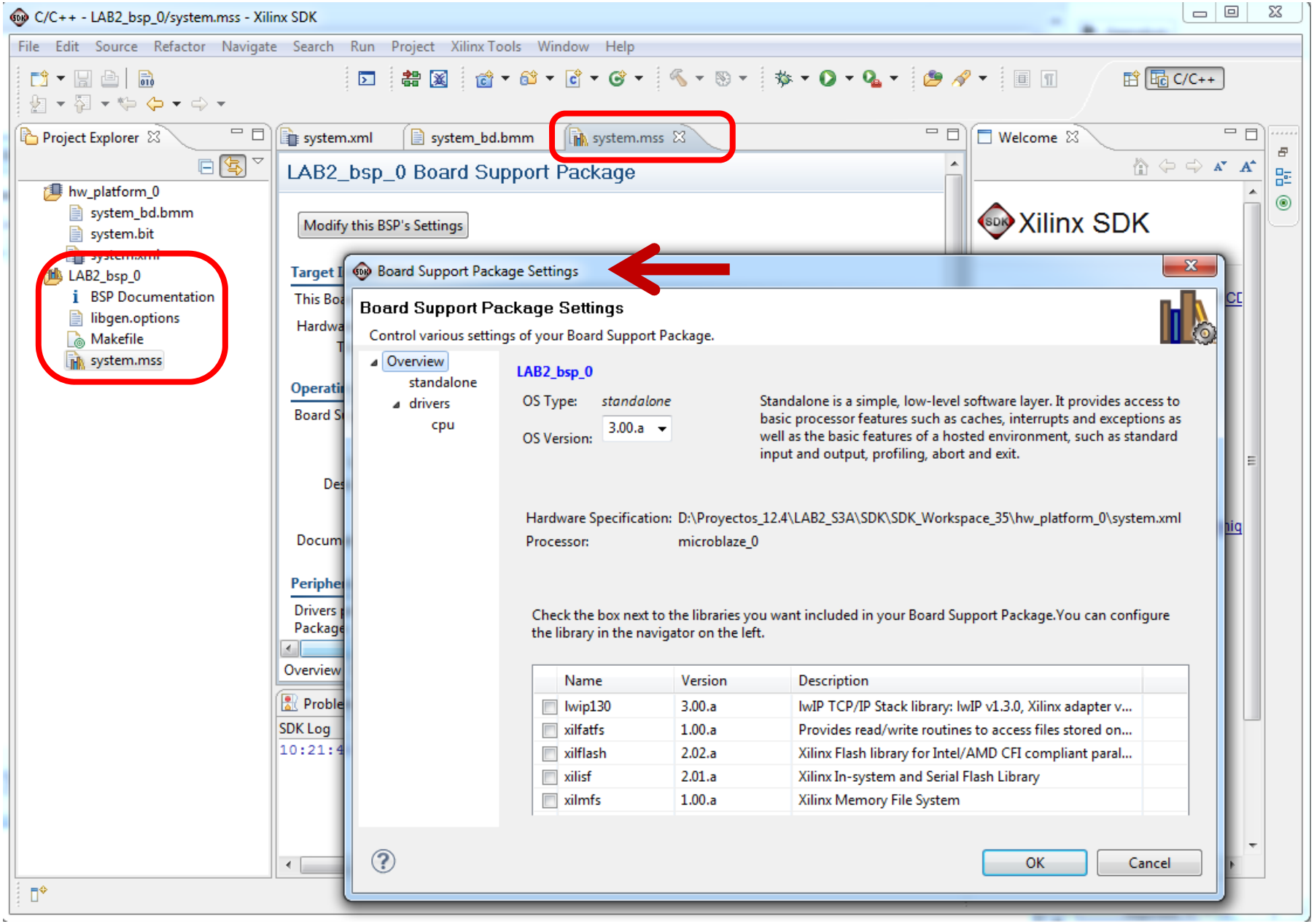
Board Support Package (BSP)

- Configuración de la plataforma SW
- Colección de bibliotecas y drivers que forman la capa más baja de la aplicación SW
  - Sistema operativo
  - Bibliotecas
  - Drivers de periféricos
  - E/S estándar
  - Periférico de depuración
- Equivalente a MSS
- Al salvarlo, ejecuta *LibGen*
- Crea carpeta en el Workspace
- Pueden existir varios BSP para un mismo HW

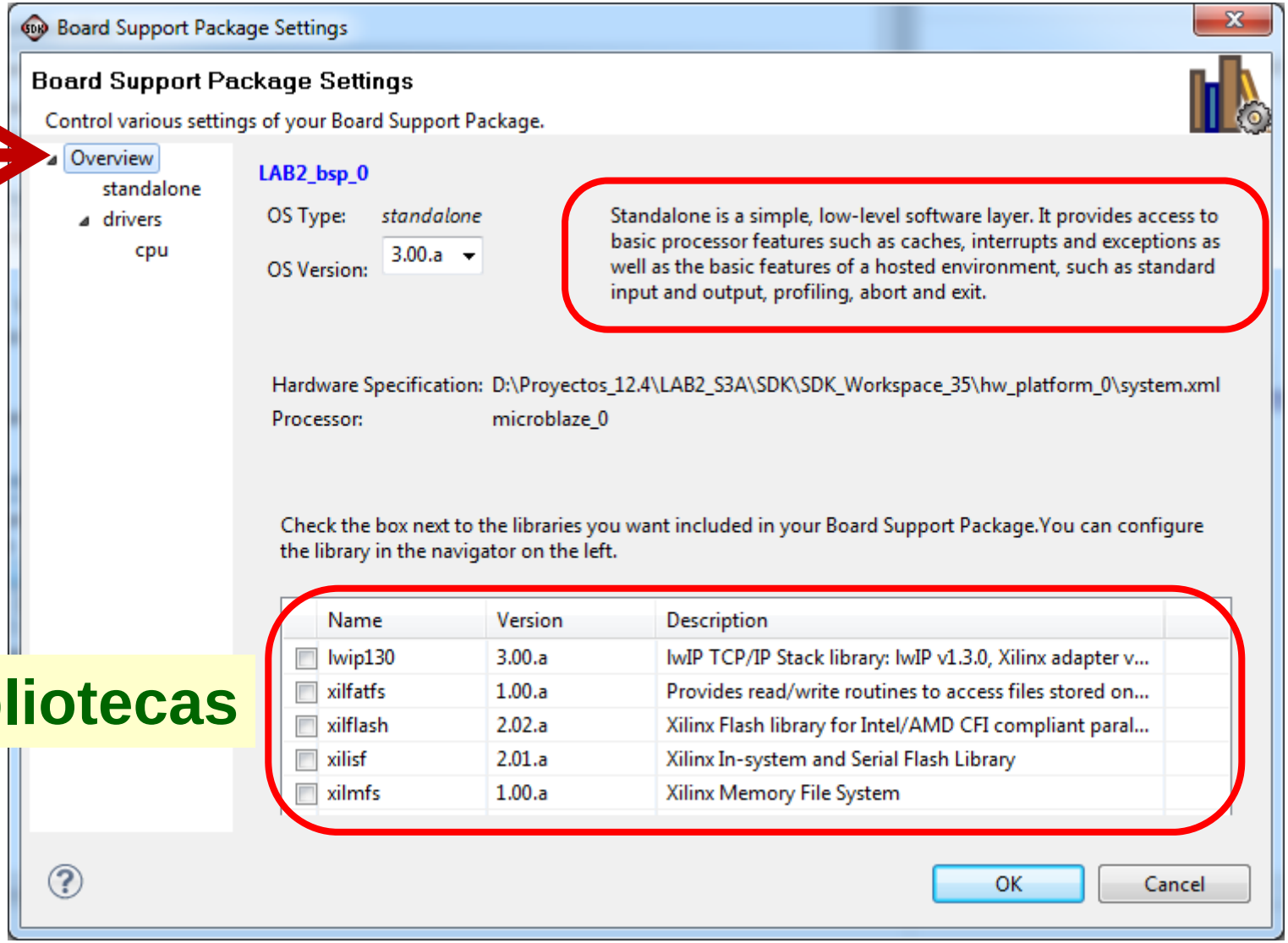
3a. Crear BSP


New Board Support Package Project
Xilinx Board Support Package Project
Create a Board Support Package.
Project name: LAB2_bsp_0
Use default location
Location: D:\Proyectos_12.4\LAB2_S3A\SDK\SDK_Workspace_35\LAB2_bsp_0
Browse...
Hardware Platform: hw_platform_0
CPU: microblaze_0
xilkernel
standalone
Standalone is a simple, low-level software layer. It provides access to basic processor features such as caches, interrupts and exceptions as well as the basic features of a hosted environment, such as standard input and output, profiling, abort and exit.
Finish
Cancel
Especificar plataforma HW
Seleccionar SO

3b. Configurar BSP

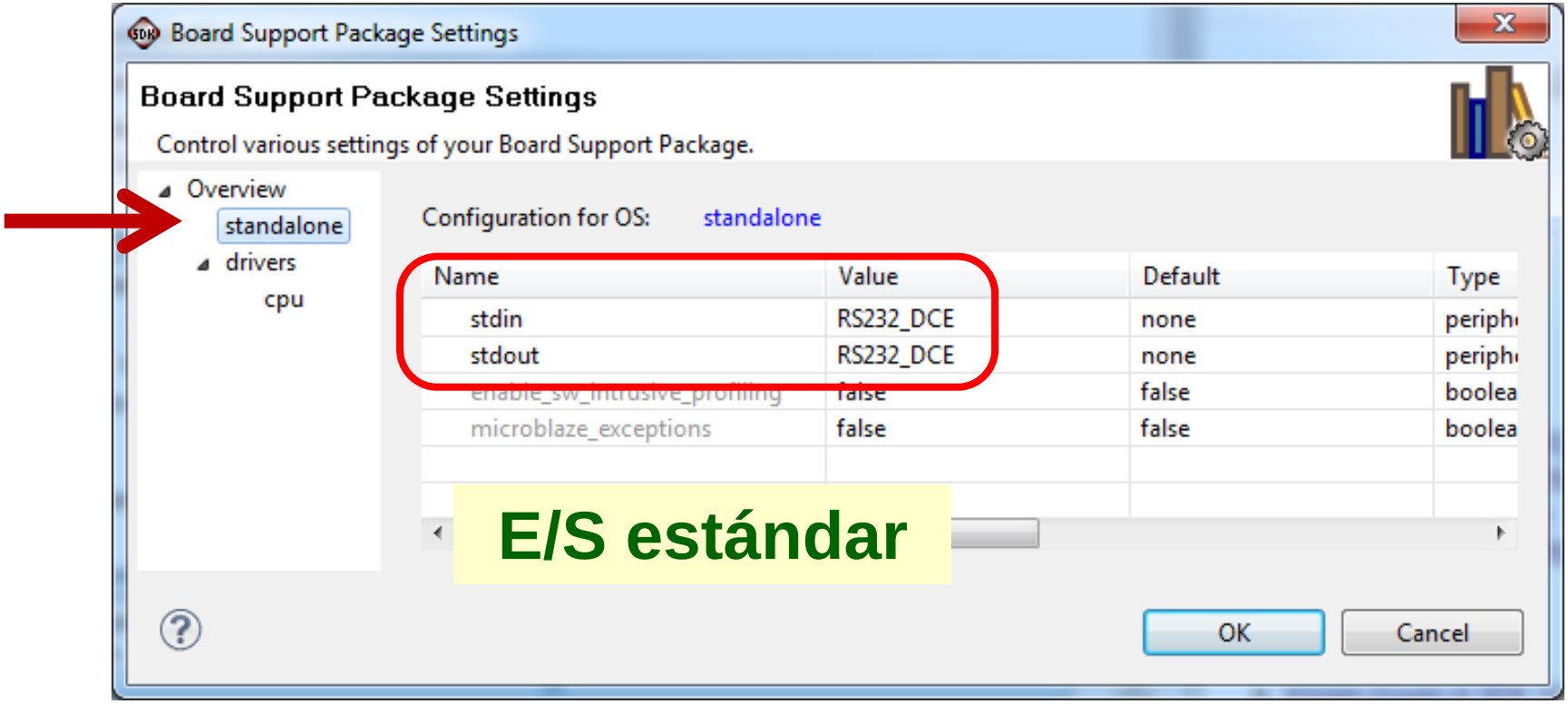


3b. Configurar BSP *(cont.)*

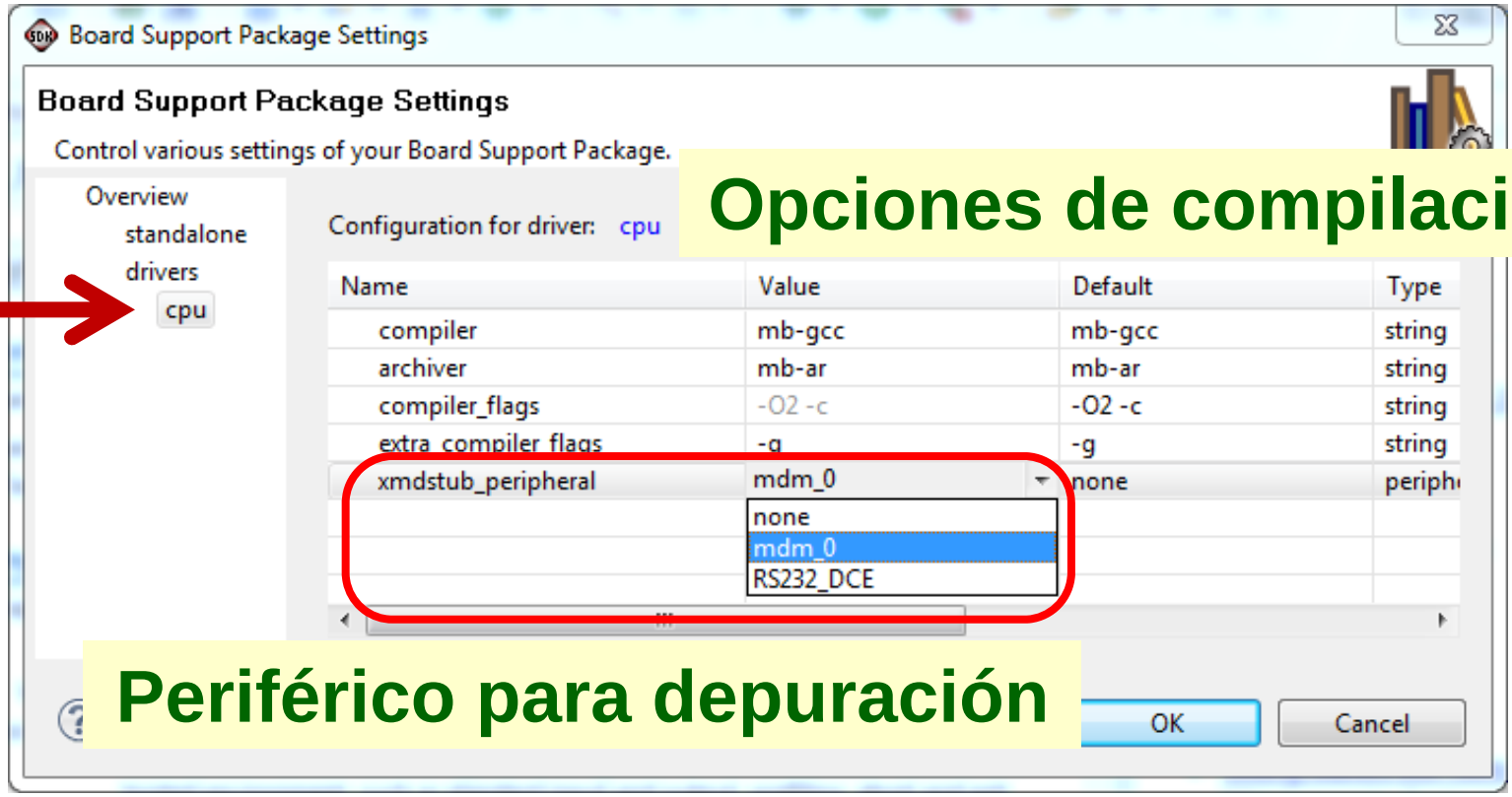


Bibliotecas

3b. Configurar BSP *(cont.)*



3b. Configurar BSP *(cont.)*

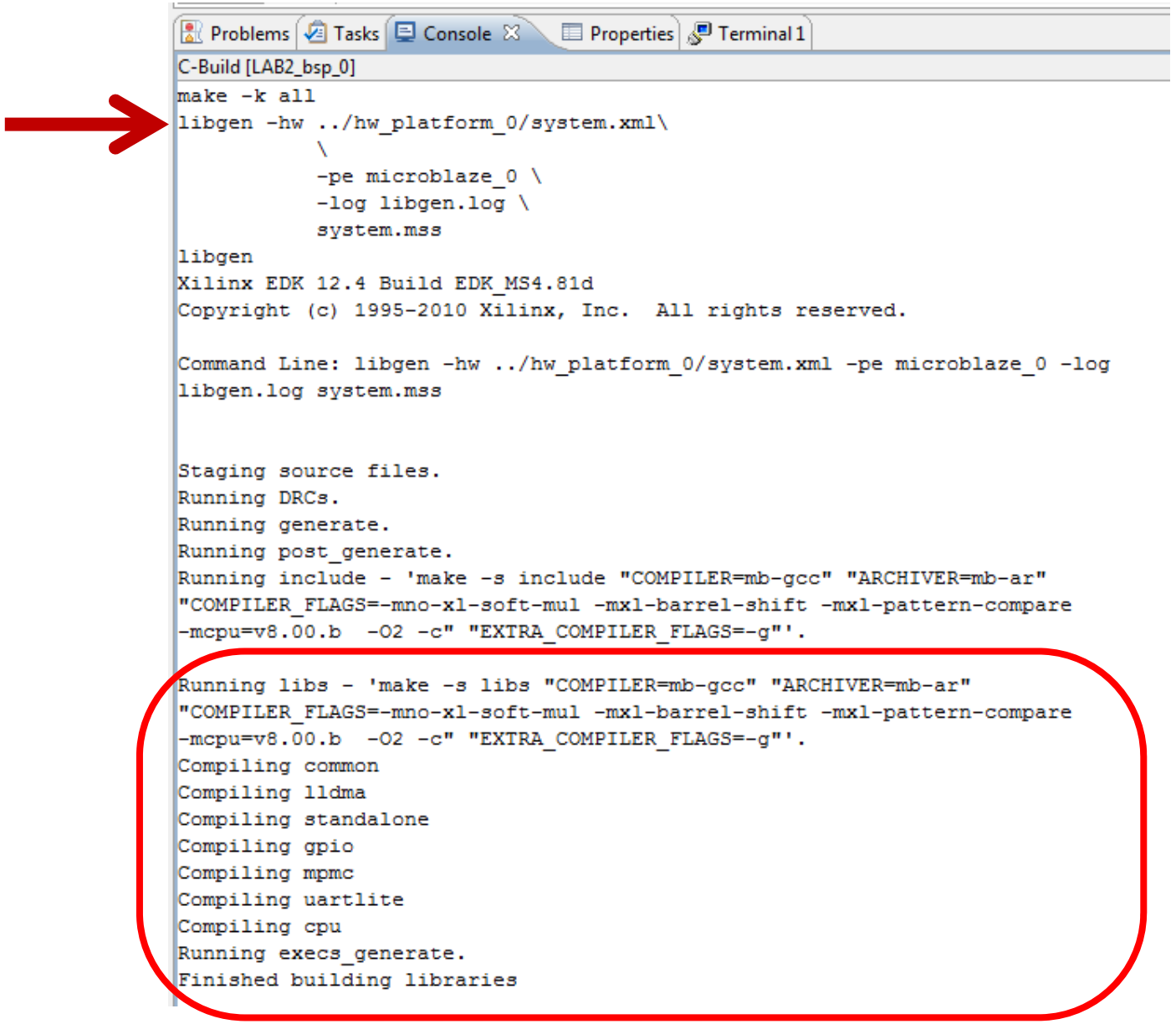


Opciones de compilación

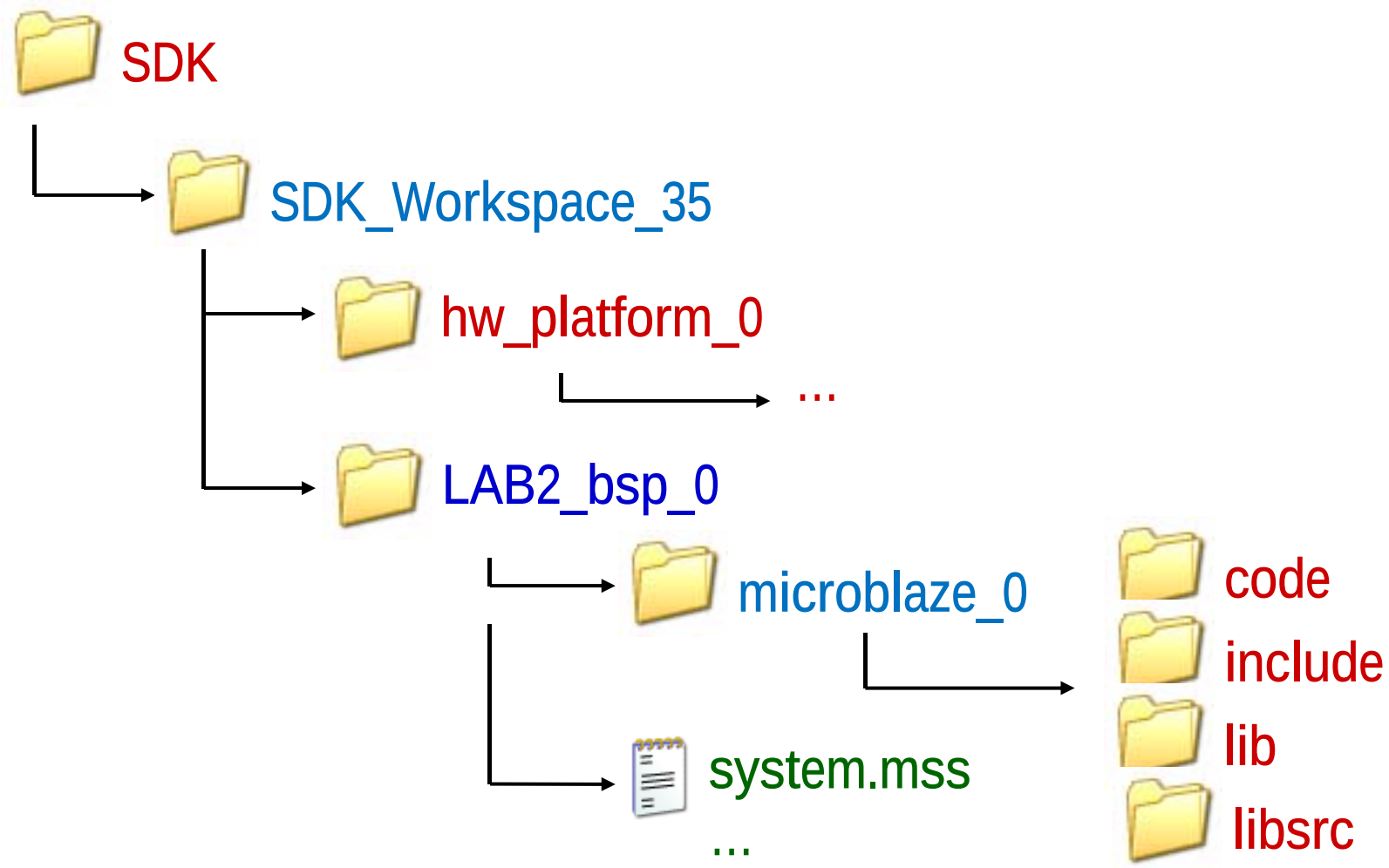
Periférico para depuración

Cada vez que finaliza la configuración SW (BSP), se ejecuta LibGen automáticamente

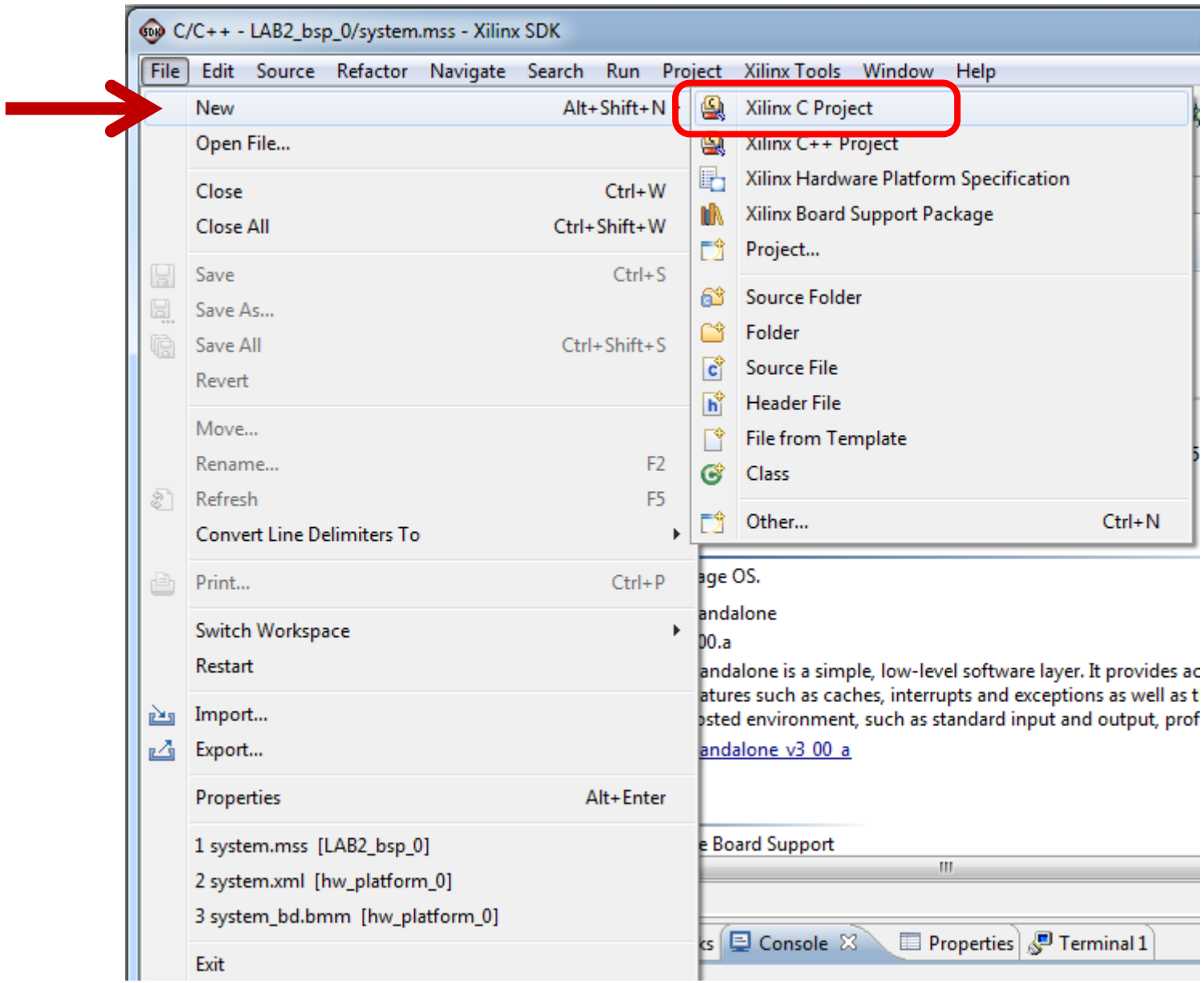
3c. Ejecución de LibGen



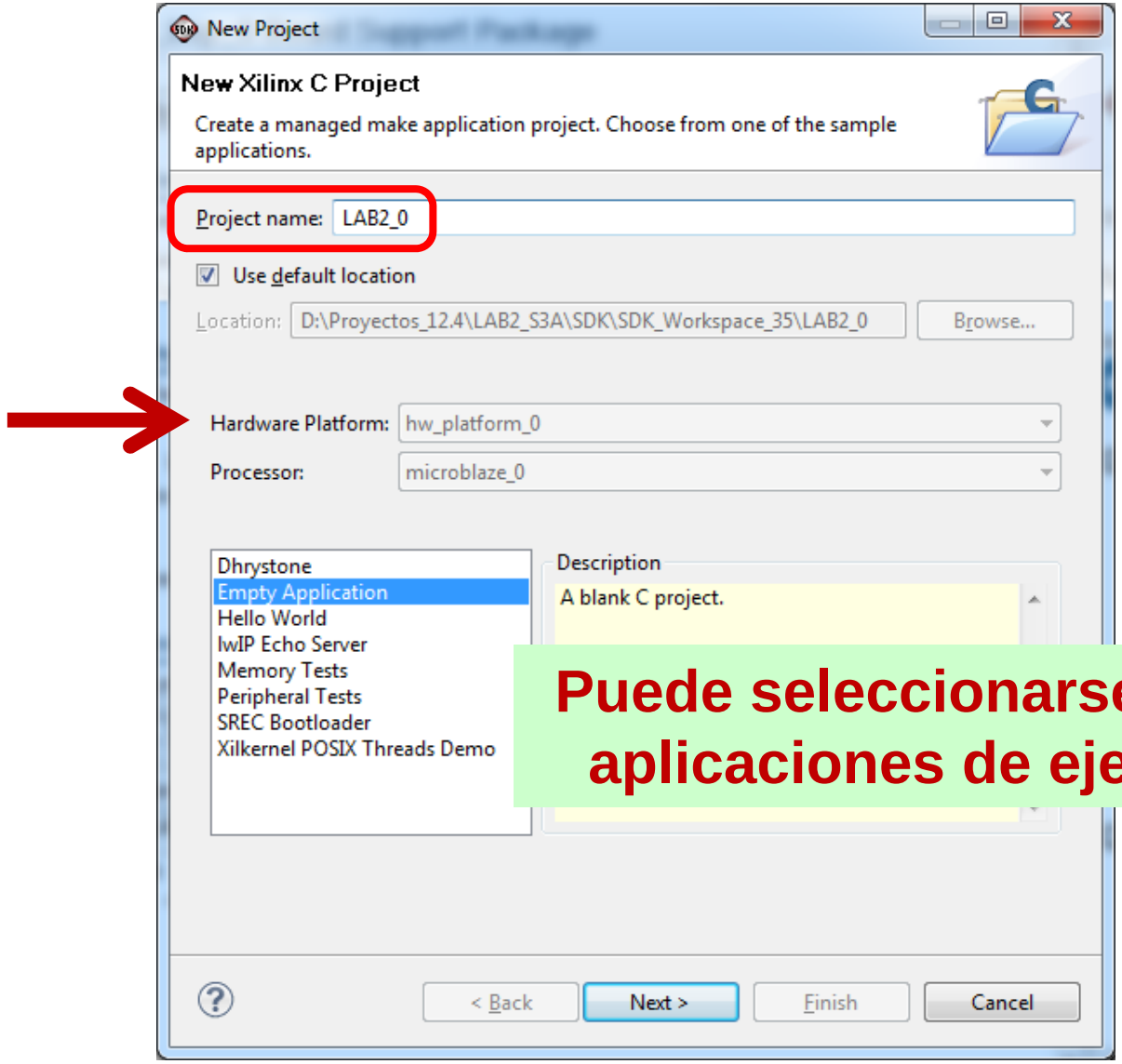
Estructura del BSP



4a. Crear proyecto SW

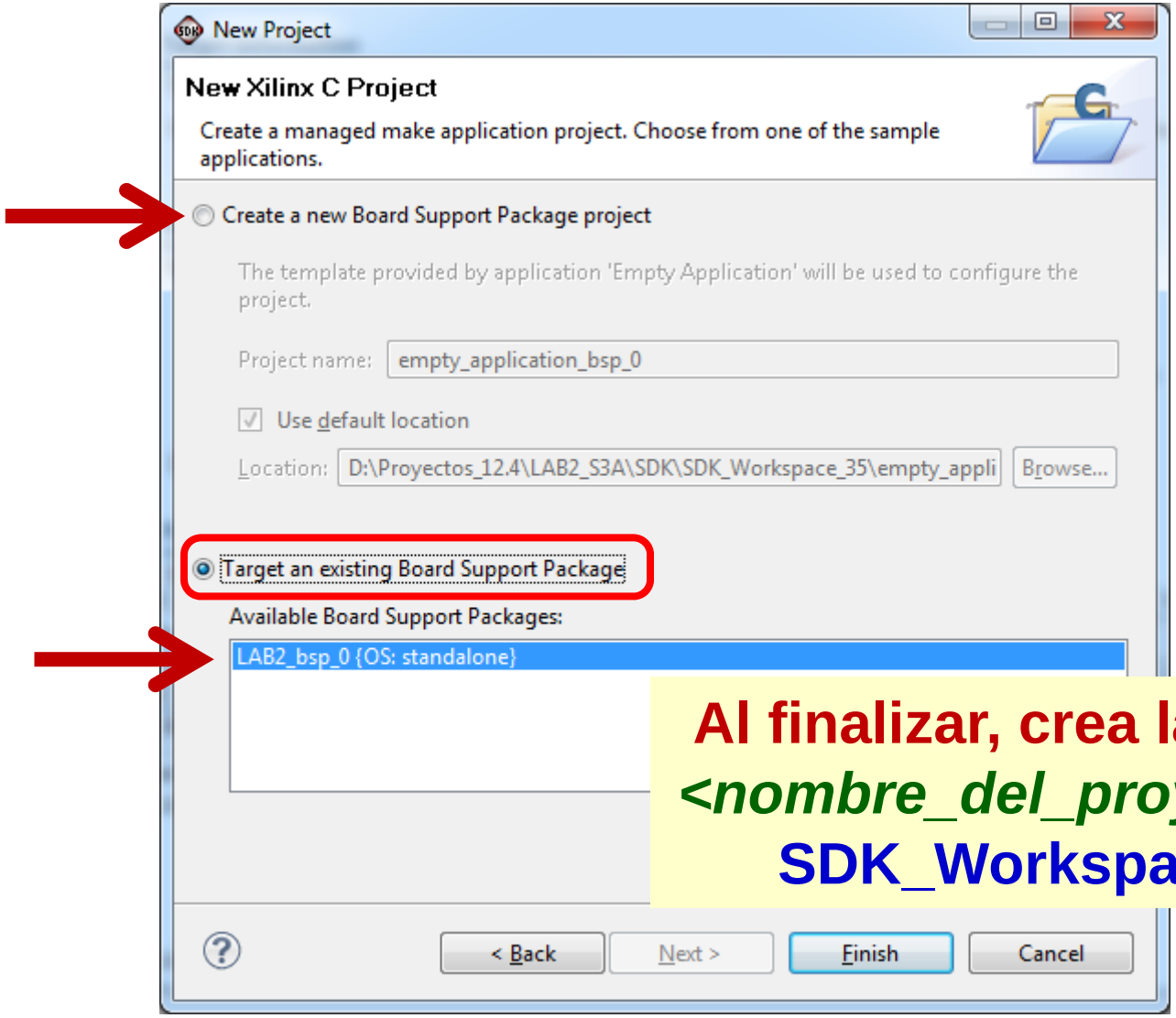


4a. Crear proyecto SW



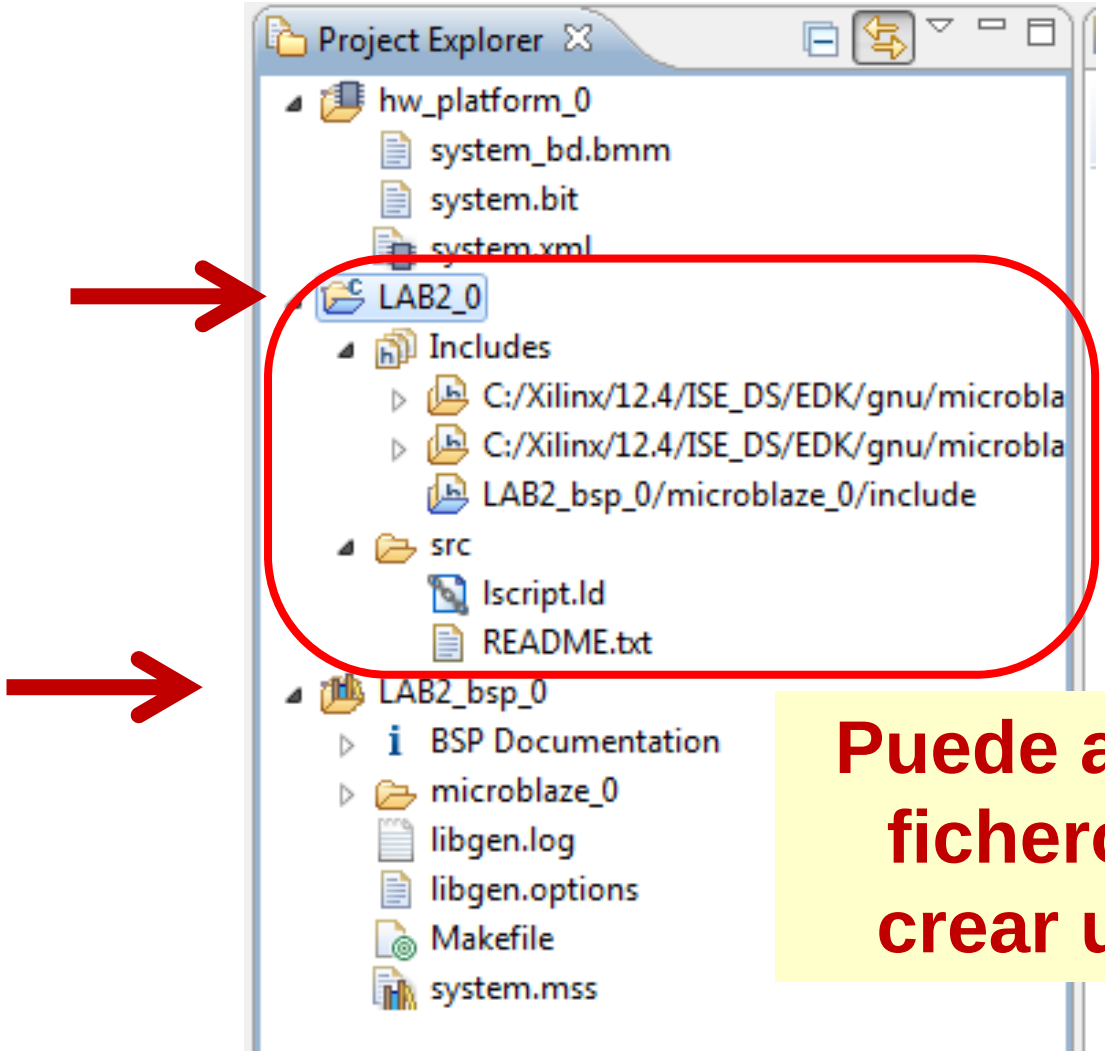
Puede seleccionarse entre aplicaciones de ejemplo

4b. Asociar con BSP



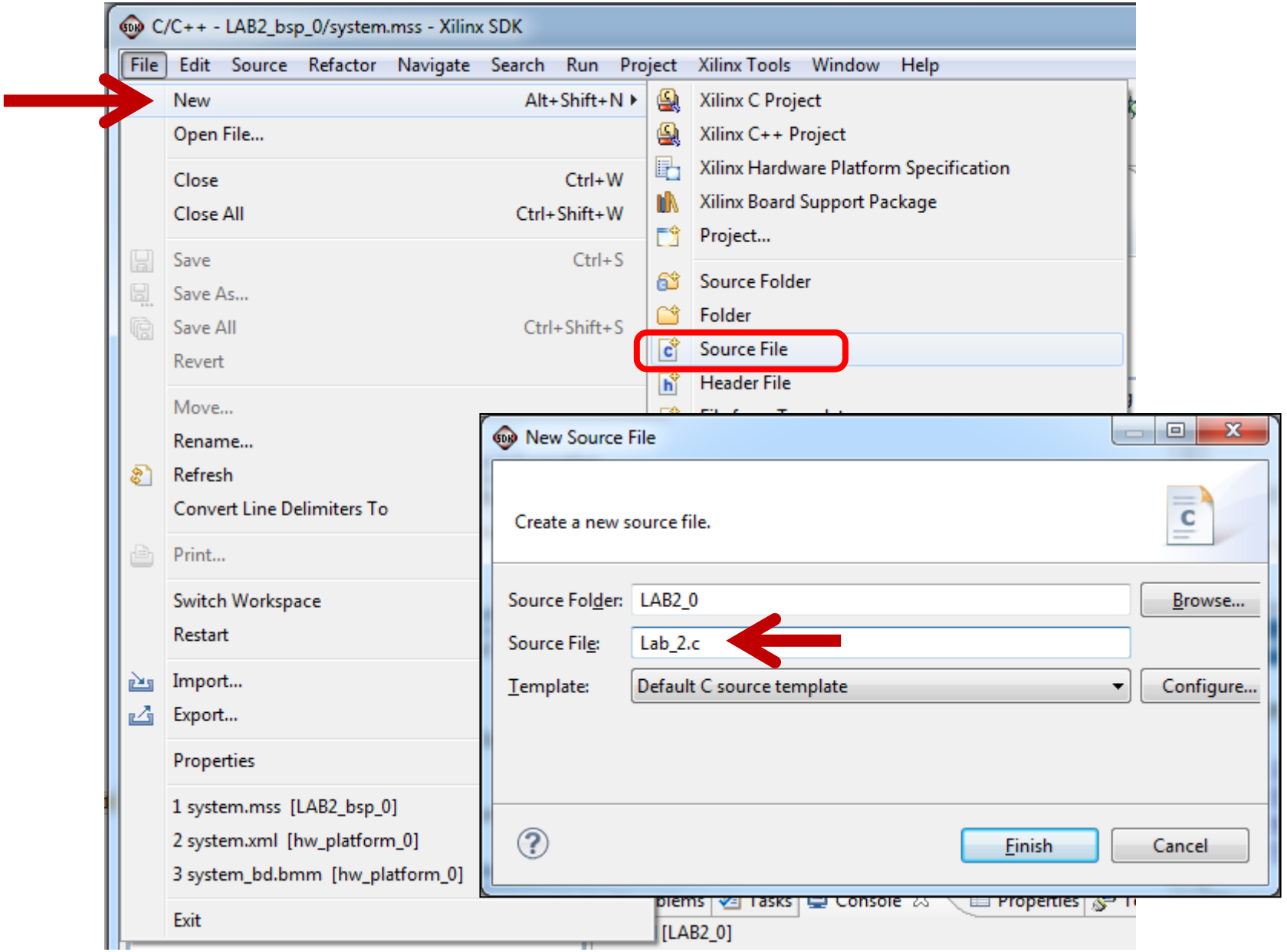
Al finalizar, crea la carpeta *<nombre_del_proyecto>* en SDK_Workspace_35

Proyecto SW

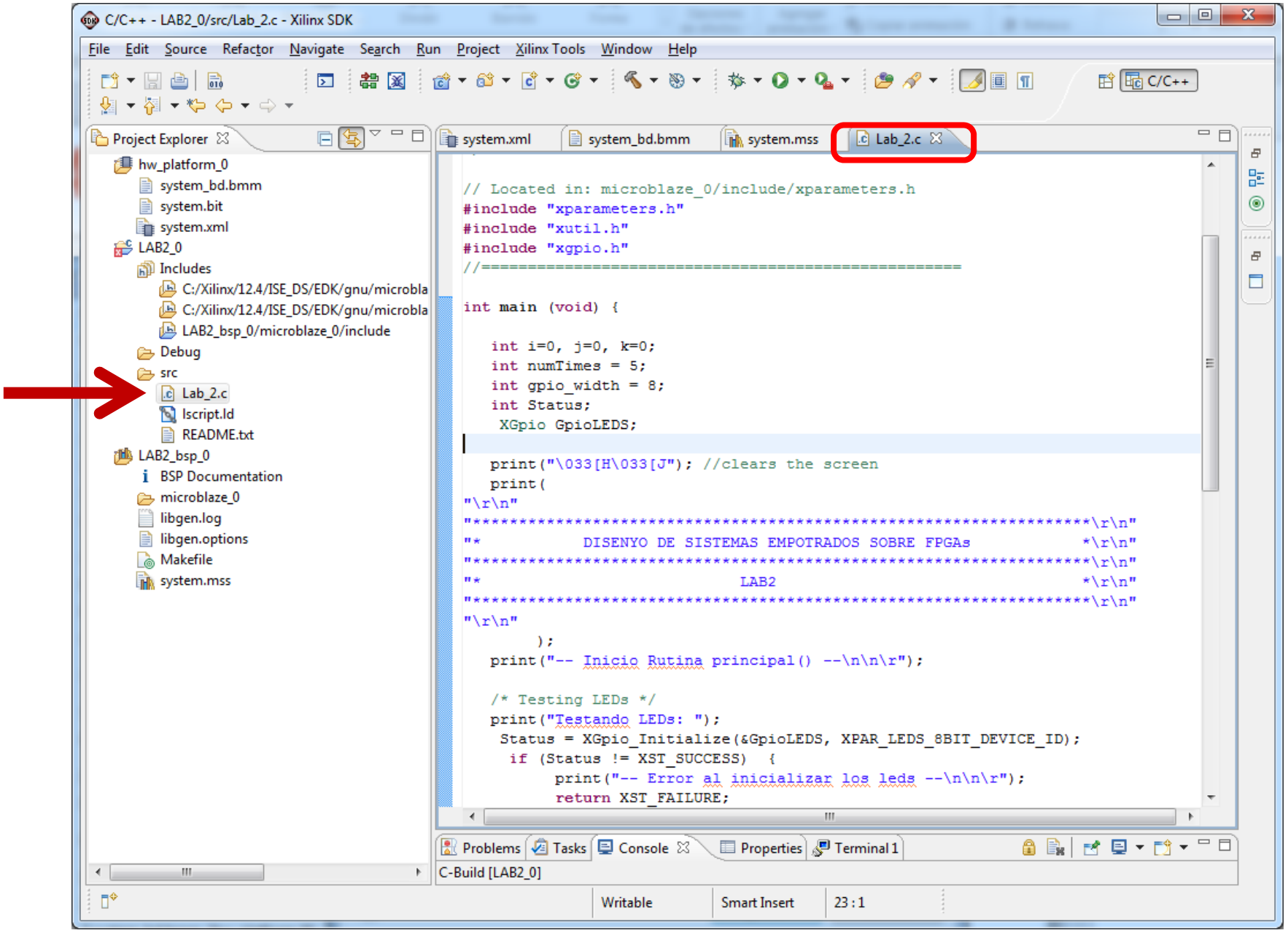


Puede añadirse un fichero fuente o crear uno nuevo

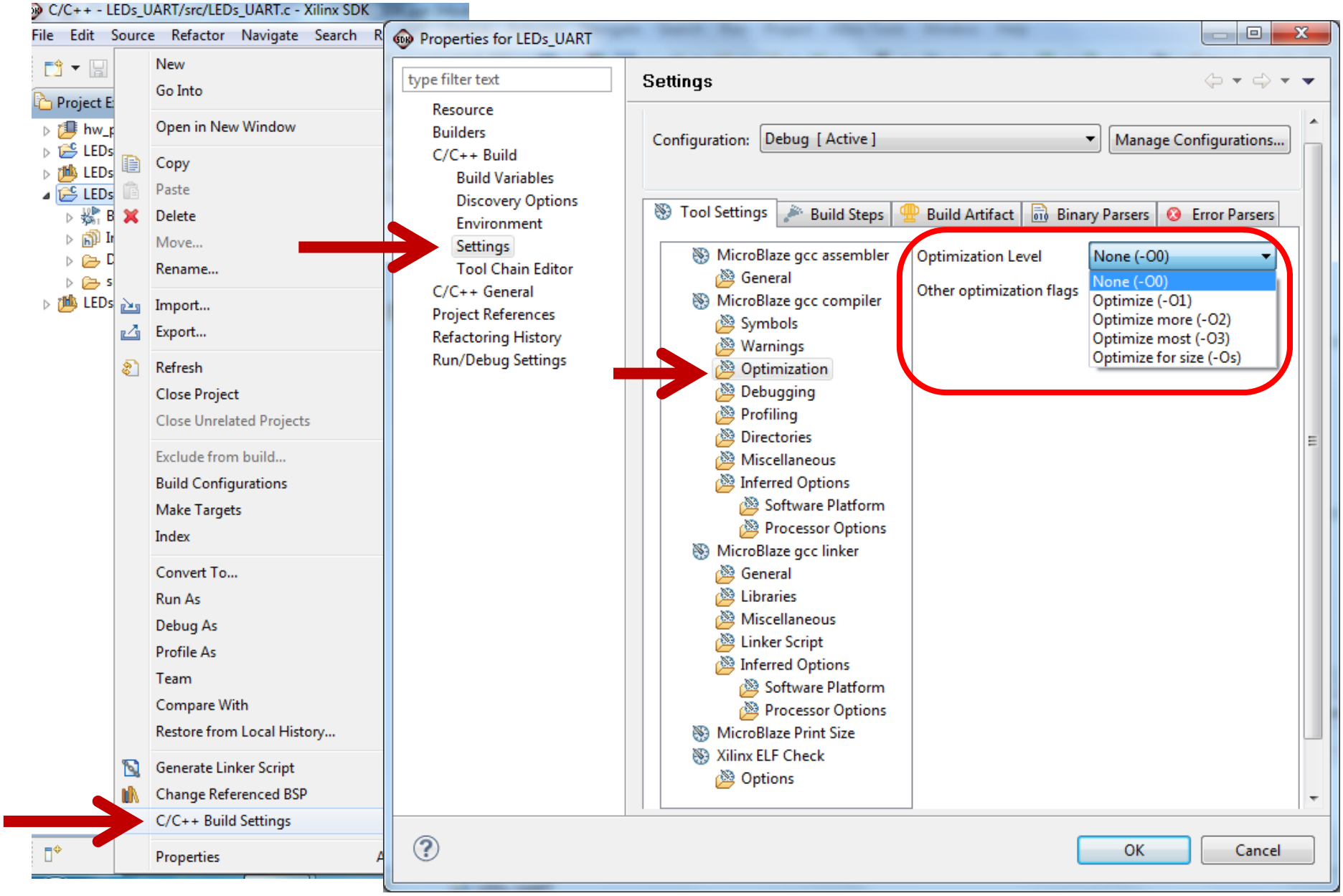
4c. Crear código fuente



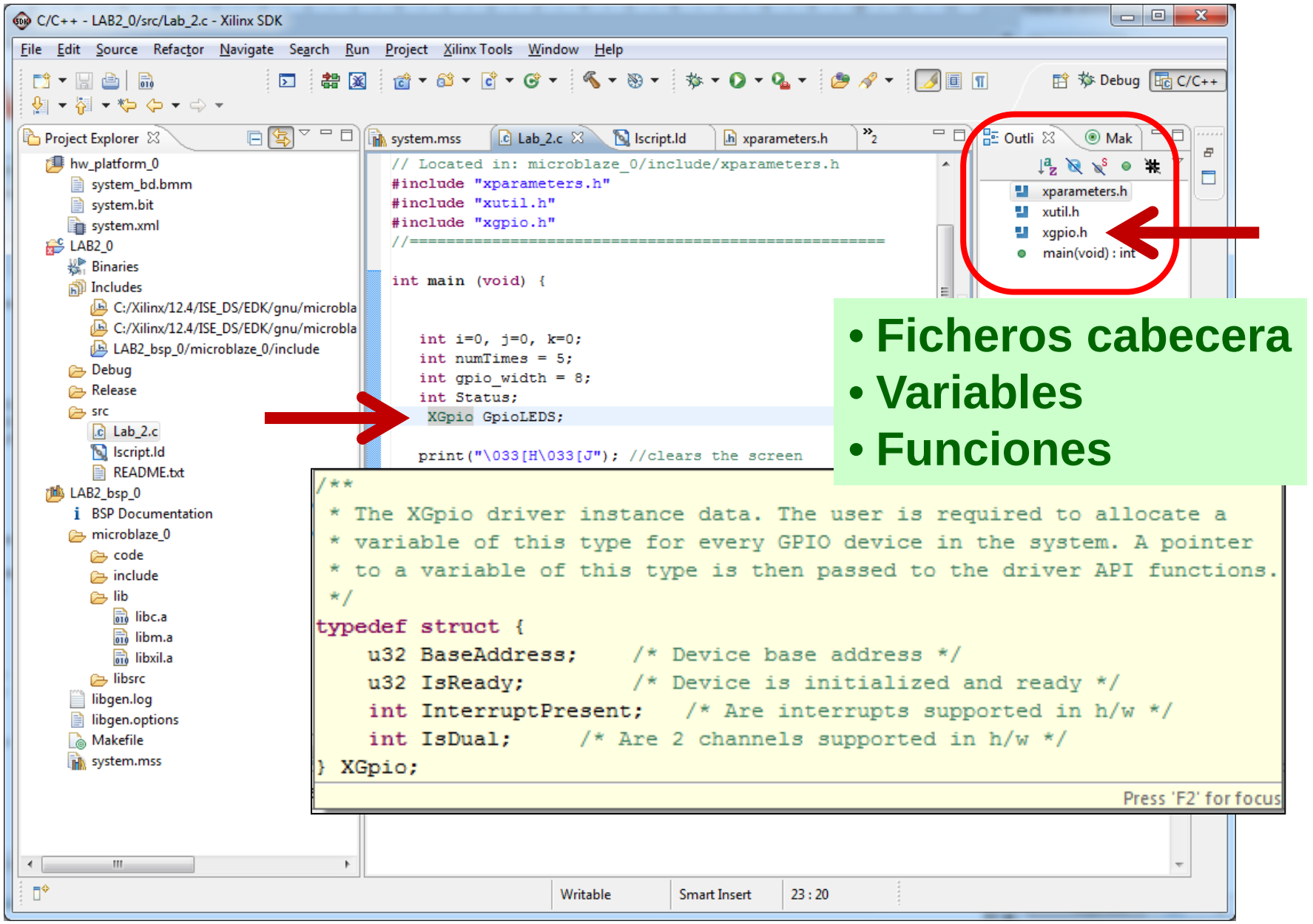
4c. Crear código fuente *(cont.)*



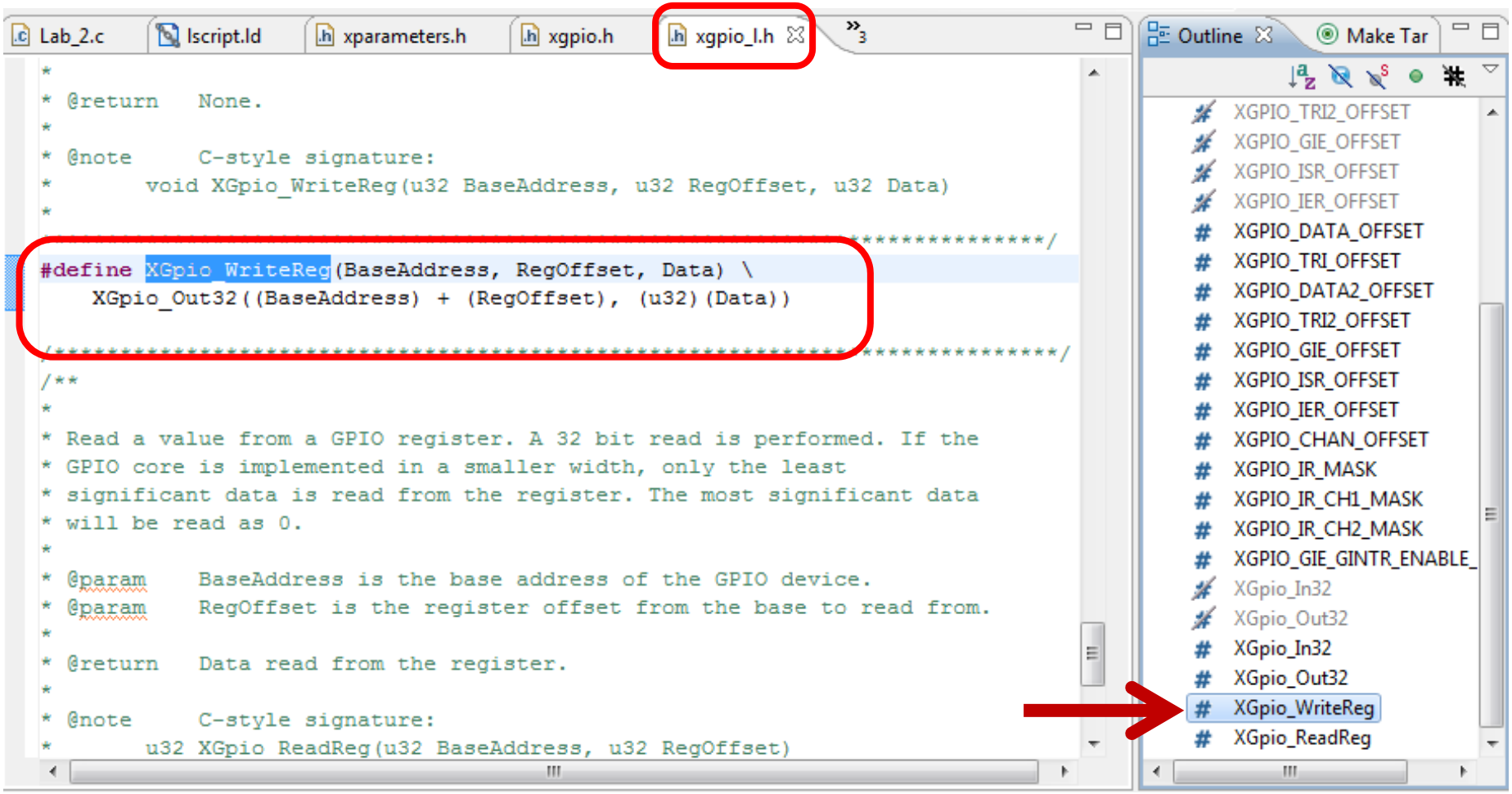
4d. Optimización de la aplicación



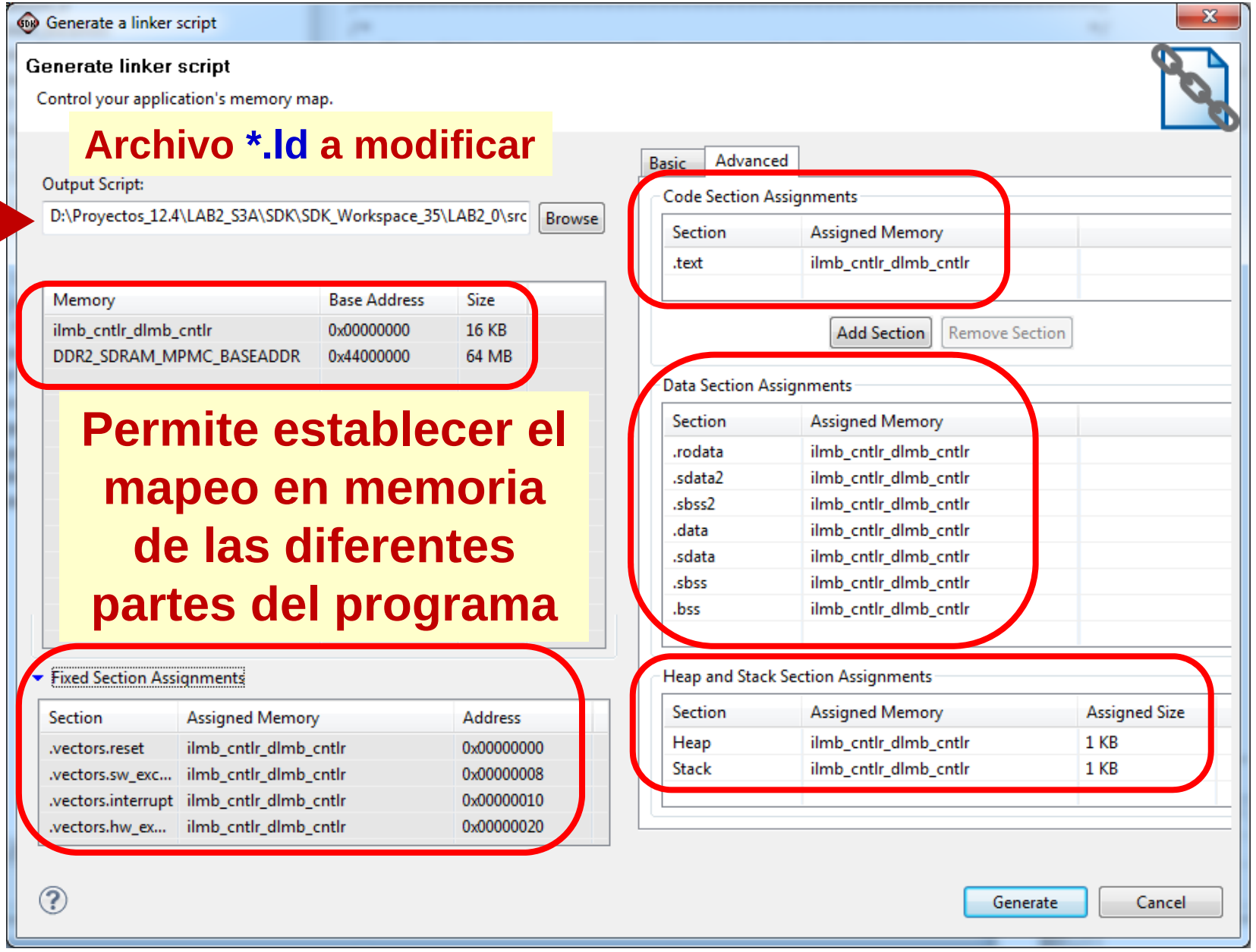
Navegador de código Eclipse



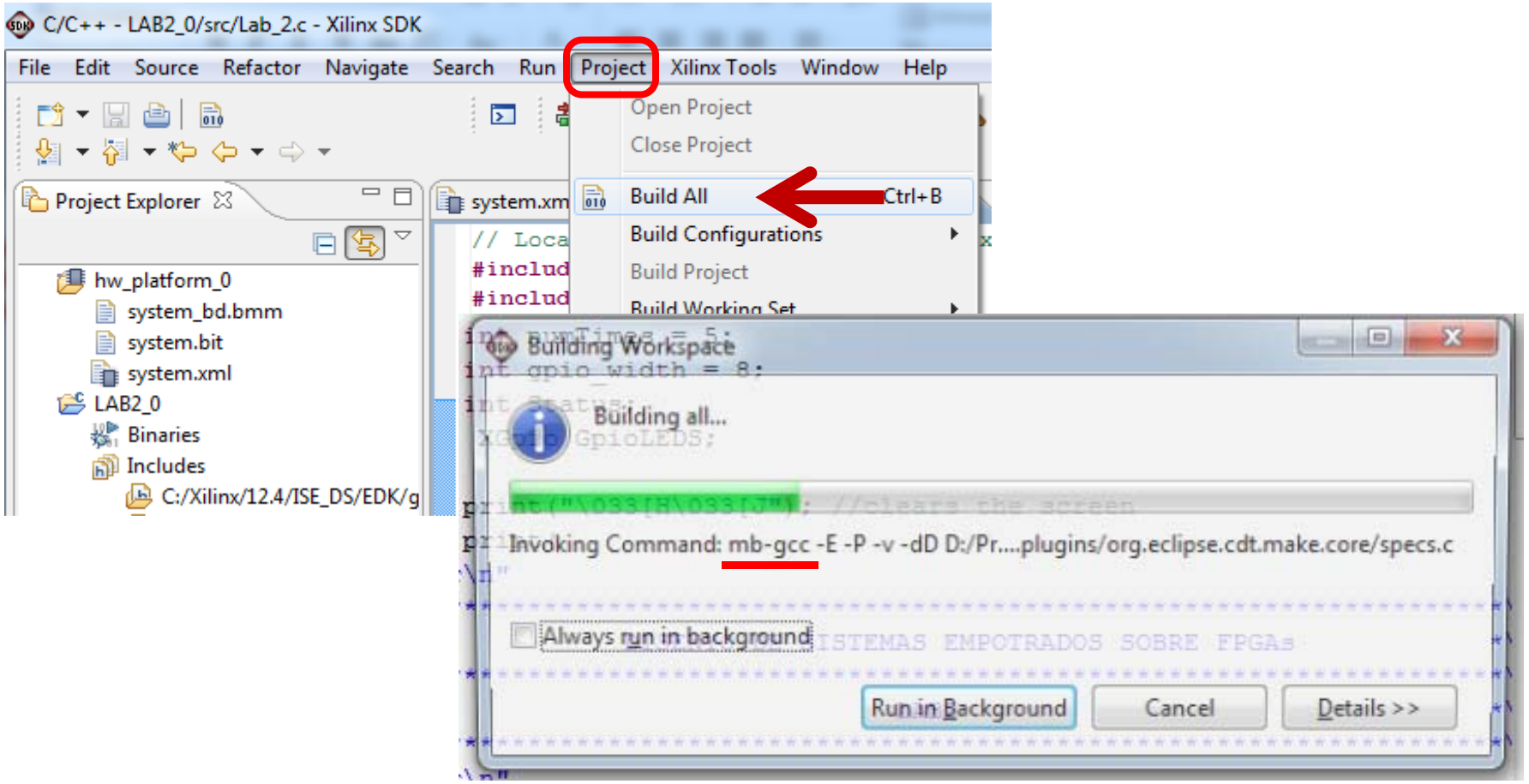
Navegador de código Eclipse



5. [Configurar LinkerScript]



6a. Implementar aplicación



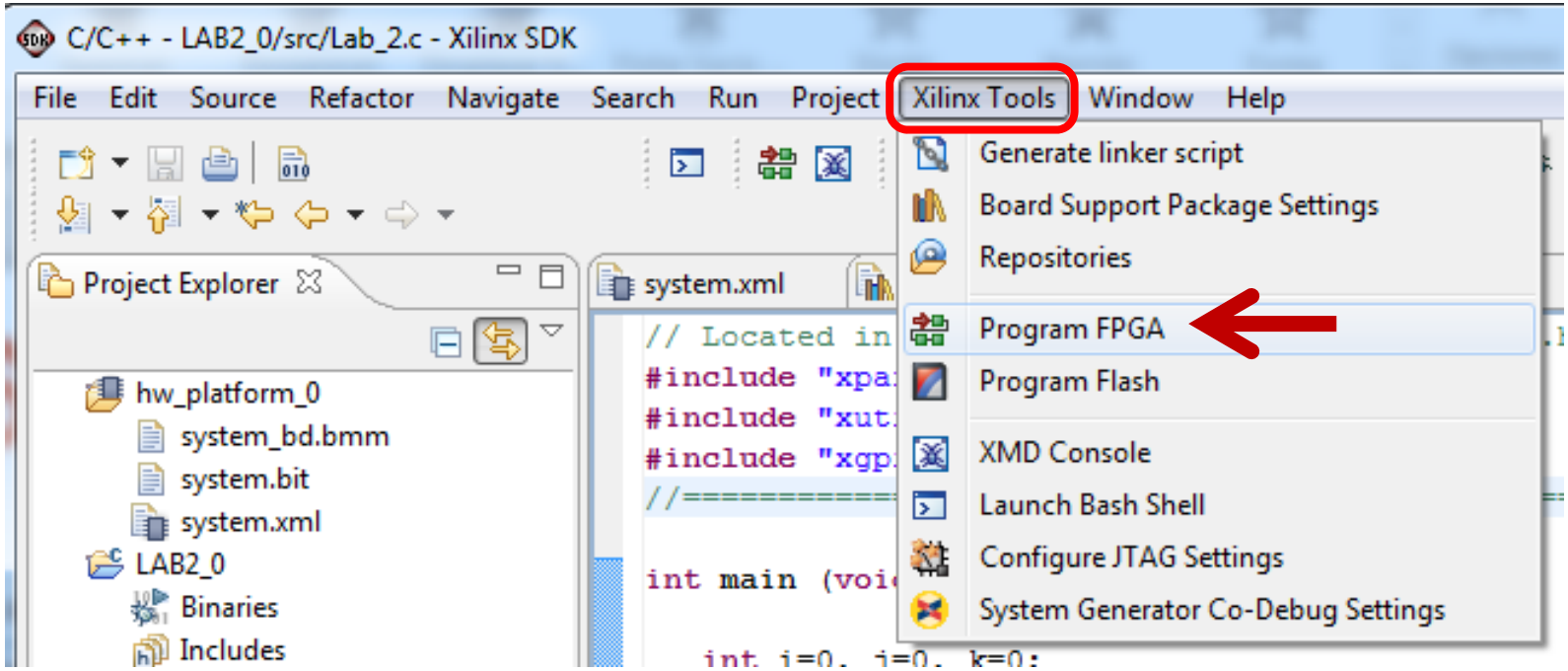
6b. Reporte de compilación

Problems | Tasks | Console | Properties | Terminal 1

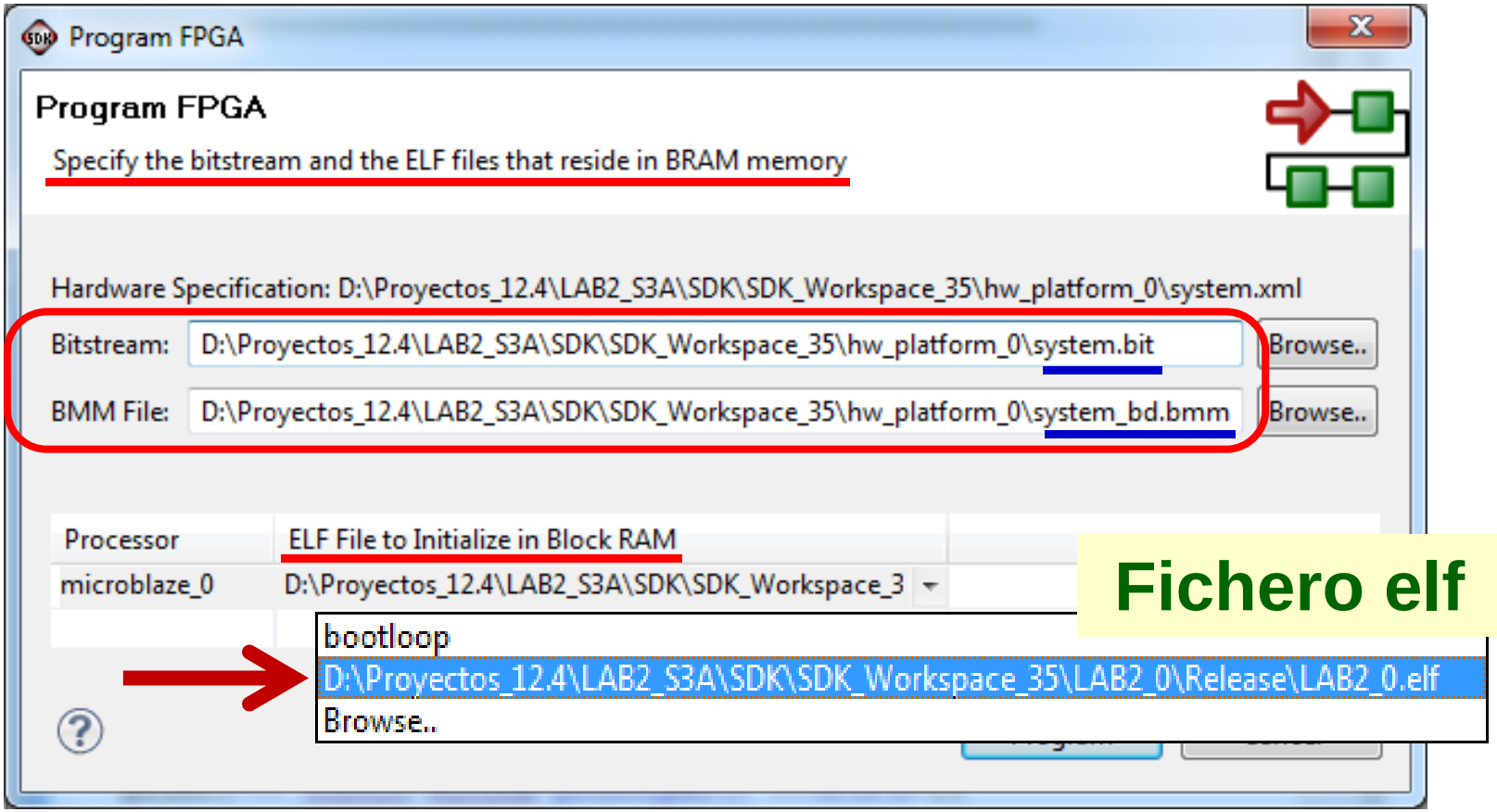
C-Build [LAB2_0]

```
Building target: LAB2_0.elf
Invoking: MicroBlaze gcc linker
mb-gcc -Wl,-T -Wl,../src/lscript.ld -L../../LAB2_bsp_0/microblaze_0/lib -mxl-barrel-shift
-mxl-pattern-compare -mcpu=v8.00.b -mno-xl-soft-mul -o"LAB2_0.elf"  ./src/Lab_2.o
Finished building target: LAB2_0.elf

Invoking: MicroBlaze Print Size
mb-size LAB2_0.elf  |tee "LAB2_0.elf.size"
   text    data     bss     dec     hex filename
   3646     348    2098    6092    17cc LAB2_0.elf
Finished building: LAB2_0.elf.size

Invoking: Xilinx ELF Check
elfcheck LAB2_0.elf -hw ../../hw_platform_0/system.xml -pe microblaze_0  |tee "LAB2_0.elf.elfcheck"
elfcheck
Xilinx EDK 12.4 Build EDK_MS4.81d
Copyright (c) 1995-2010 Xilinx, Inc.  All rights reserved.

Command Line: elfcheck -hw ../../hw_platform_0/system.xml -pe microblaze_0
LAB2_0.elf

ELF file    : LAB2_0.elf
elfcheck passed.
Finished building: LAB2_0.elf.elfcheck
```

7a. Programar FPGA



7b. Generar download.bit



Fichero elf

Se inicializa con *bootlop* cuando la aplicación residirá en memoria externa (o se descargará con XMD)

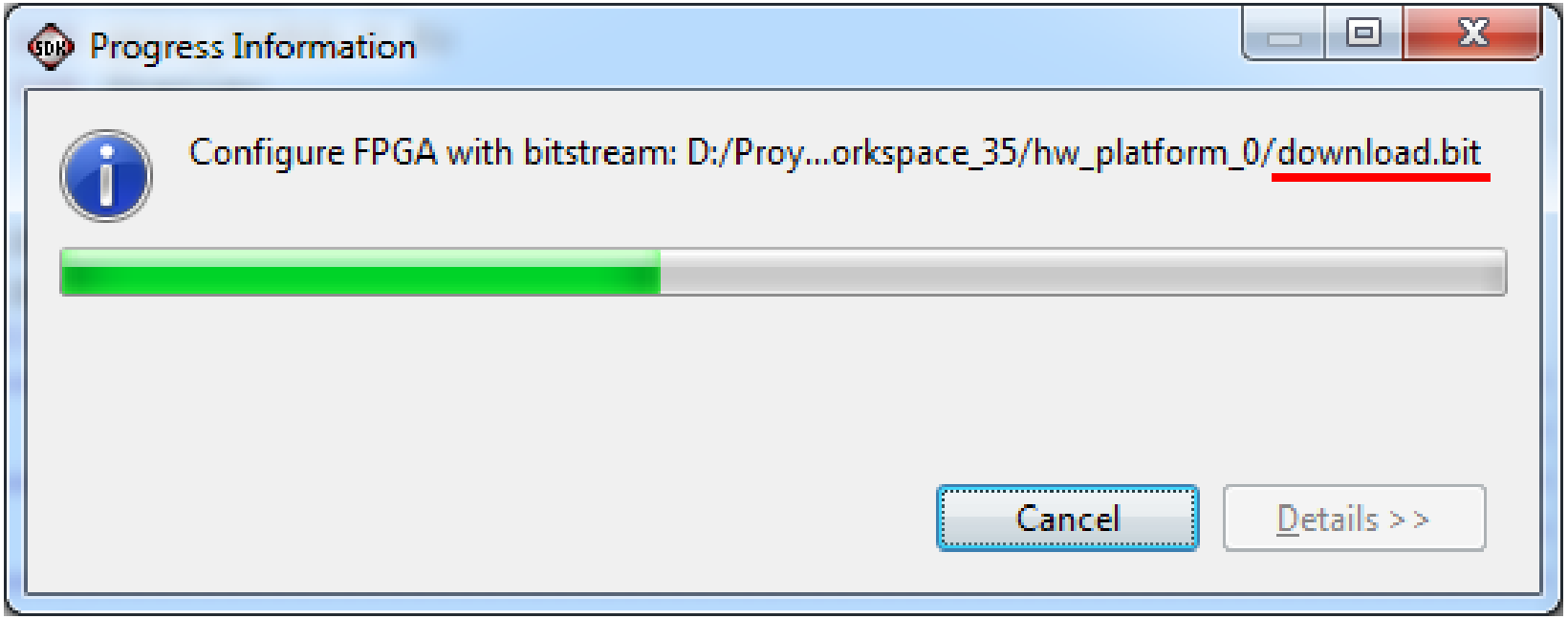
7b. Generar download.bit *(cont.)*

Problems | Tasks | Console | Properties | Terminal 1

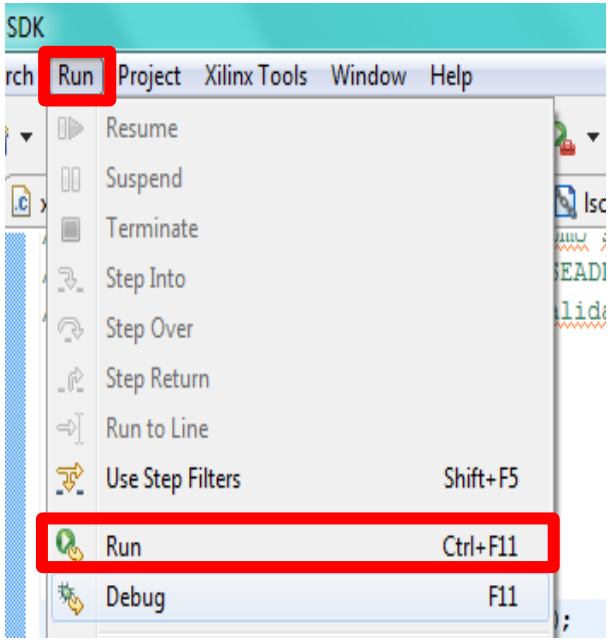
Program FPGA

```
elfcheck -hw D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system.xml \
-mode bootload -mem BRAM -pe microblaze_0 \
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/LAB2_0/Release/LAB2_0.elf
elfcheck
Xilinx EDK 12.4 Build EDK_MS4.81d
Copyright (c) 1995-2010 Xilinx, Inc.  All rights reserved.

Command Line: elfcheck -hw
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system.xml -mode
bootload -mem BRAM -pe microblaze_0
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/LAB2_0/Release/LAB2_0.elf

ELF file        :
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/LAB2_0/Release/LAB2_0.elf
elfcheck passed.

data2mem -bm D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system_bd.bmm \
-bt D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/system.bit -bd \
D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/LAB2_0/Release/LAB2_0.elf tag microblaze_0 \
-o b D:/Proyectos_12.4/LAB2_S3A/SDK/SDK_Workspace_35/hw_platform_0/download.bit
```

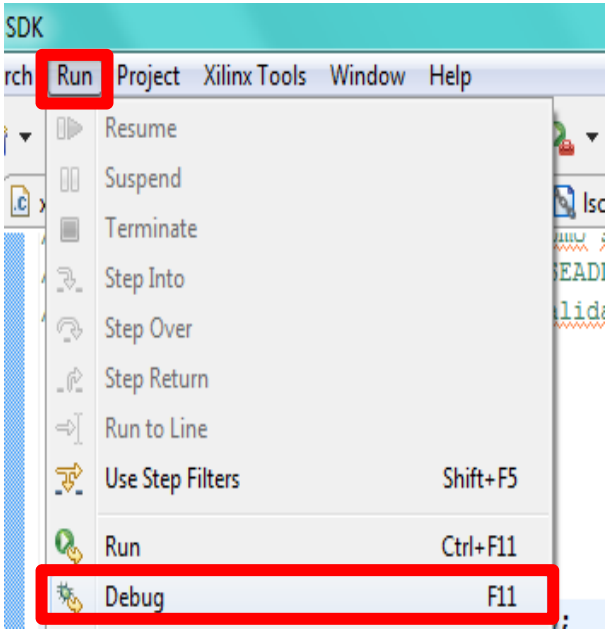
7c. Descargar bitstream al FPGA



8a. Depuración sobre el FPGA



Run

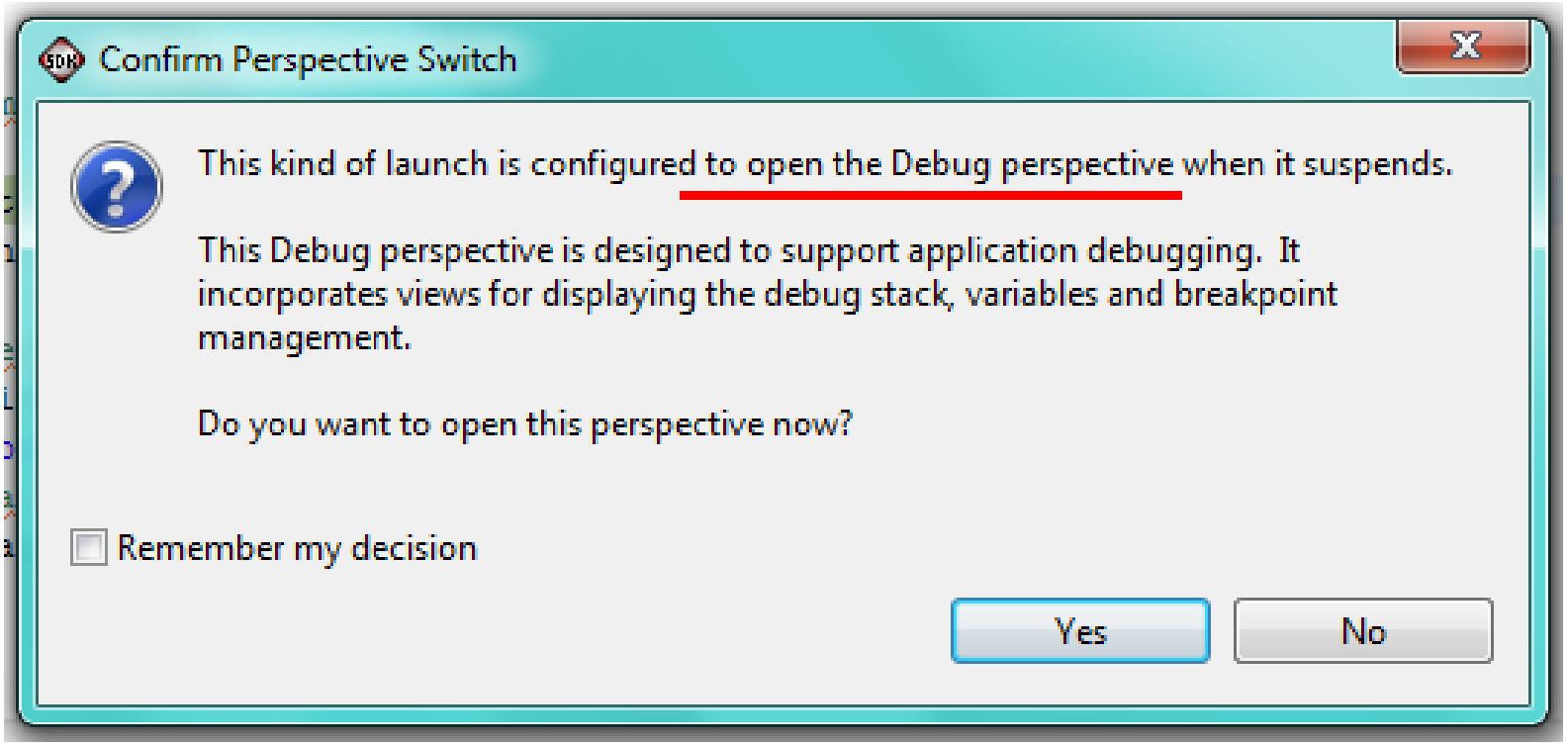


Debug

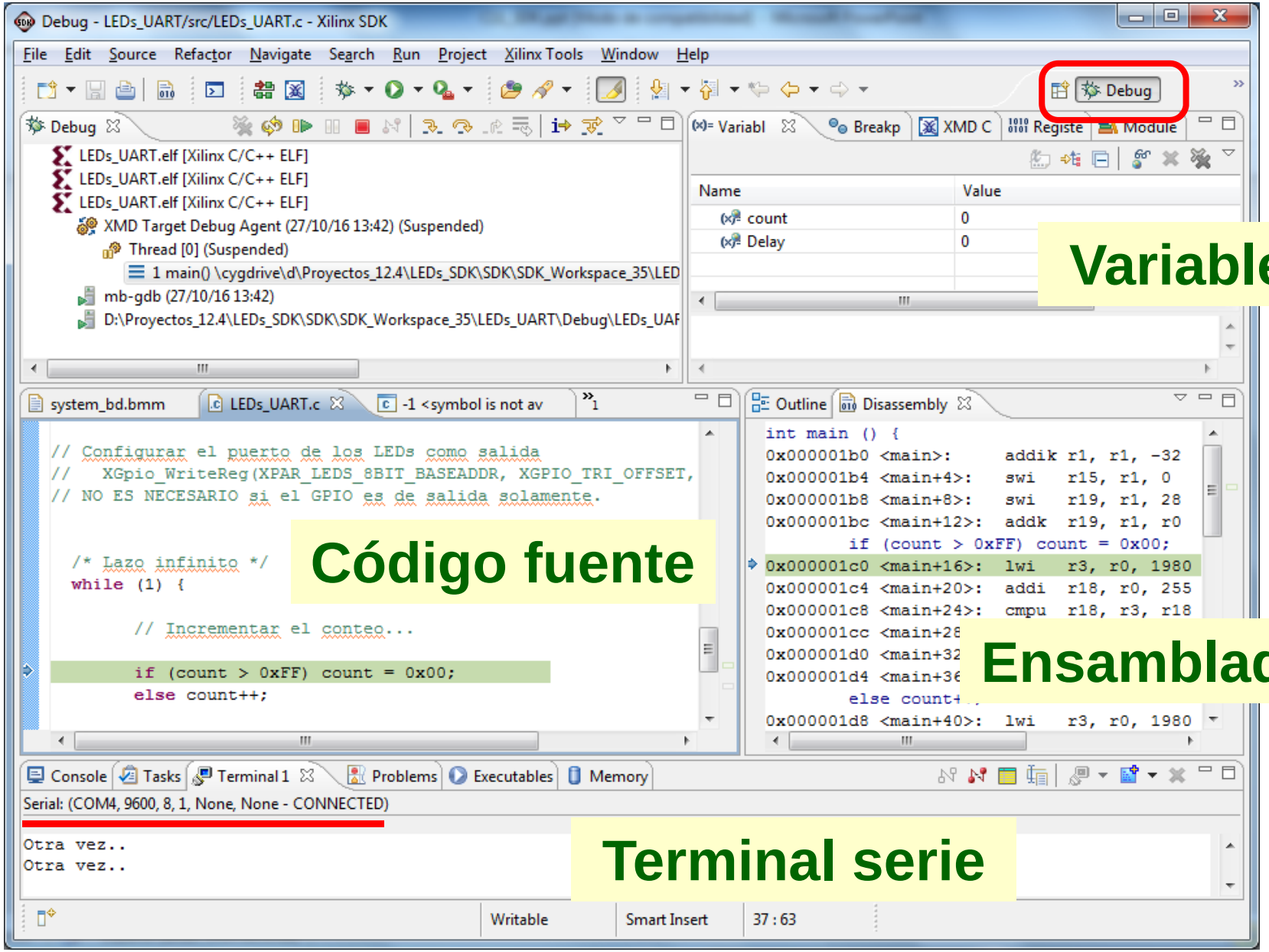
Permite correr la aplicación y detenerla desde SDK

Permite depurar la aplicación desde SDK, el cual traduce cualquier acción de la interfaz de usuario a comandos GDB, y procesa las respuestas del GDB para mostrarlos en pantalla

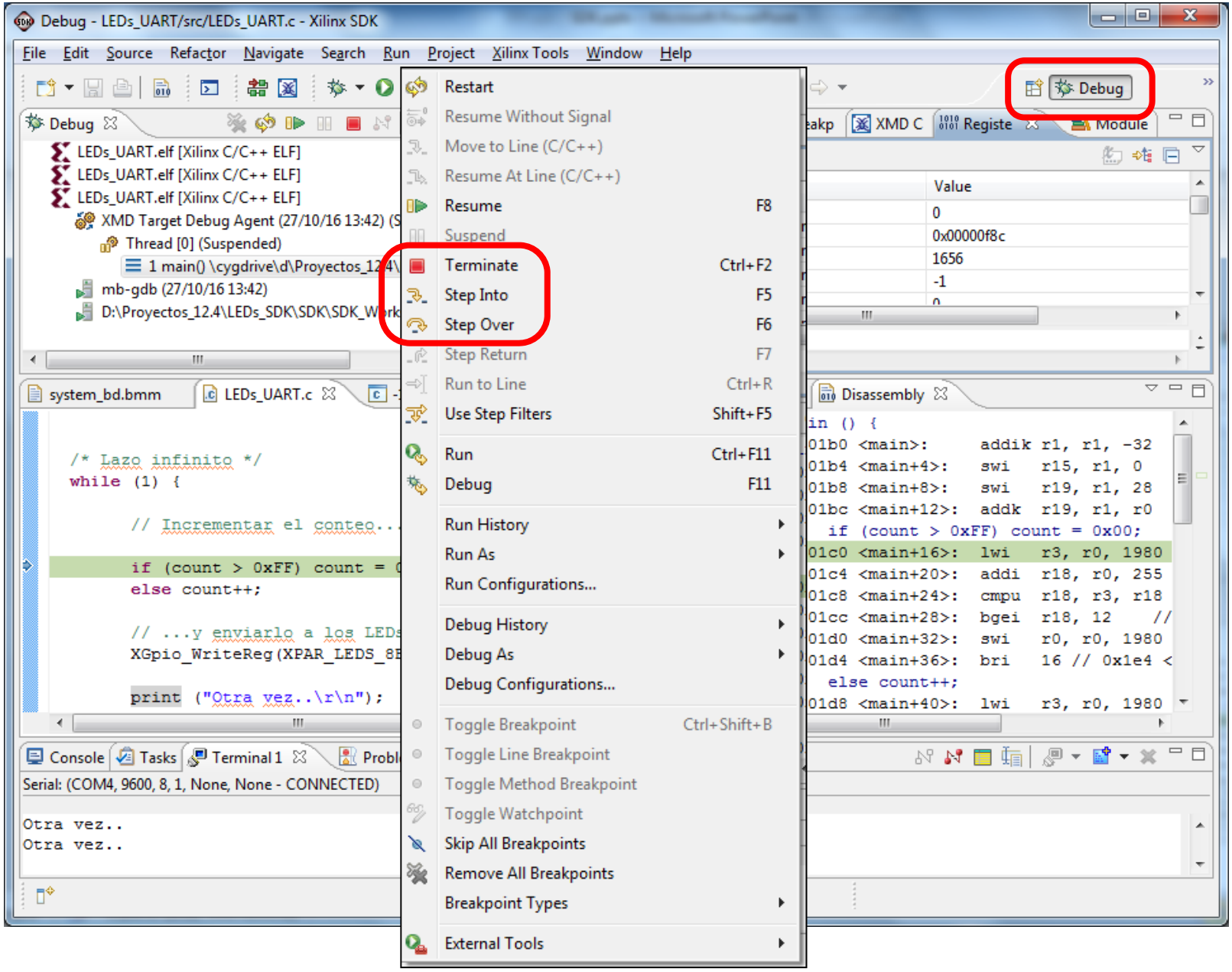
8a. Depuración sobre el FPGA



8b. Perspectiva de depuración



8c. Comandos de depuración



Documentación

- **Manuales**
  - ***Getting started with Xilinx SDK***
  - ***SDK Cheat Sheet Tutorials***

- **Soporte Web**
  - **SDK**
    - **http://www.support.xilinx.com/sdk**