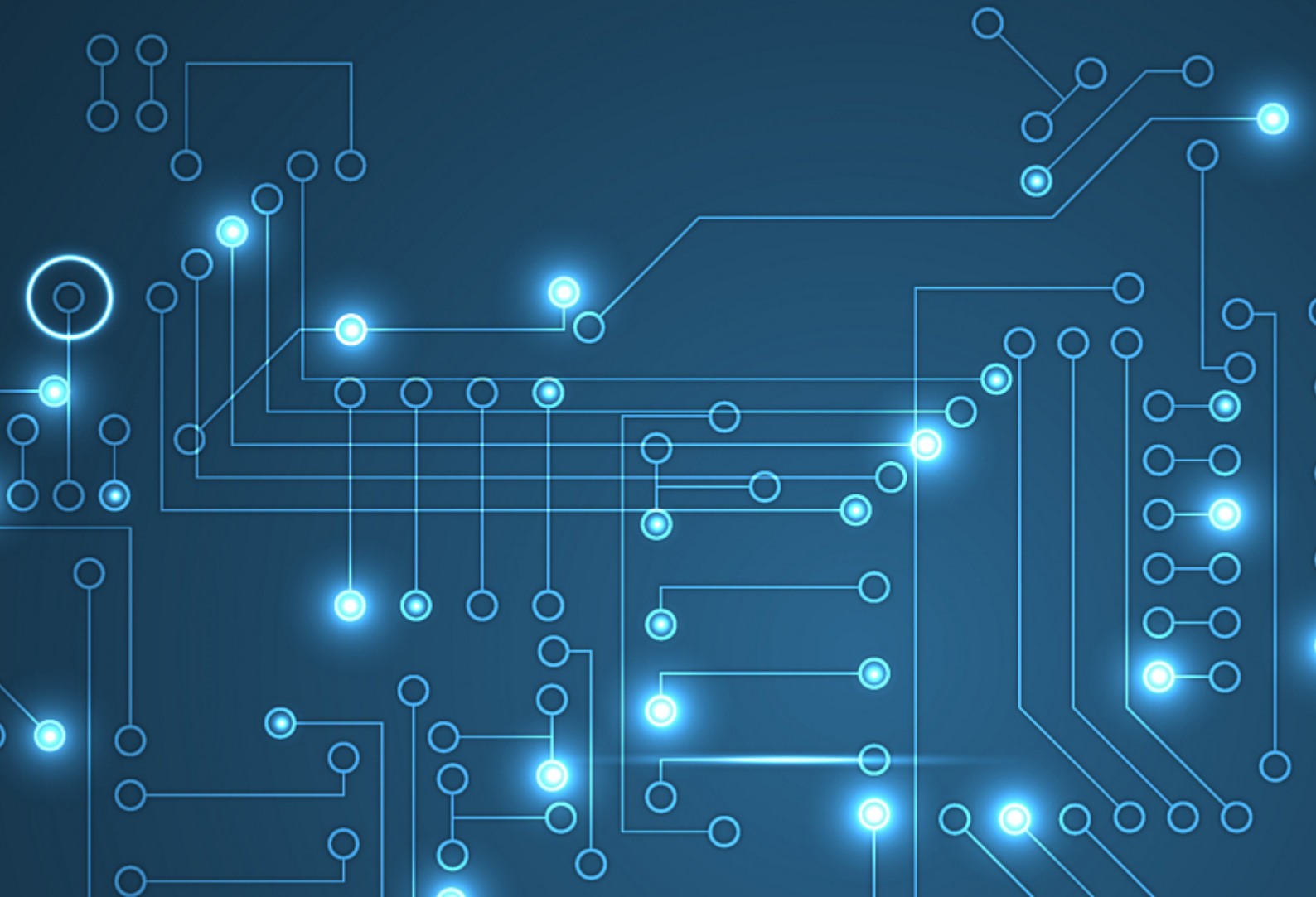
A companion guide to the text book: A Practical Introduction to the Xilinx Zynq-7000 Adaptive SoC

Vivado and SDK Development Flow

Author: Derek Murray

Version: 0.1

Date: 10/6/21



Revision History

Version	Date	Comment
0.1	10/6/21	First draft

Table 18.1. Revision History

Limit of Liability/Disclaimer of Warranty: The author makes no representation or warranty with respect to the accuracy or completeness of the contents of this work and specifically disclaims all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. If improperly wired, circuits described in this work may possibly cause damage to the device and physical injury. The author shall not be liable for damages arising herefrom. The fact that an organisation or website is referred to in this work as a citation and/or a potential source of further information does not mean that the author endorses the information the organisation or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

1 Introduction

Chapter 6 in Book: Section 2

Software Project	Book Chapter	Section in this Guide
1	7	Section 3.1
2	8	Section 3.2
3	9	Section 3.3
4	10	Section 3.4
5	11	Section 3.5
6	12	Section 3.6
7	13	Section 3.7
8	14	Section 3.8
9	16	Section 3.9
10	18	Section 3.10

Table 18.2. Revision History

File Locations in GitHub.

If using Zybo-Z7-20:

If using ZedBoard

Possible directory structure for Zynq-7000 projects.

1.1 Sub-Heading 1

1.1.1 Sub-Sub-Heading 1

1.1.1.1 Sub-Sub-Heading 1

2 Project Design in Vivado

2.1 Create the Project

Figure 1. When Vivado first launches, select Create Project.

Figure 2. Click Next to continue.

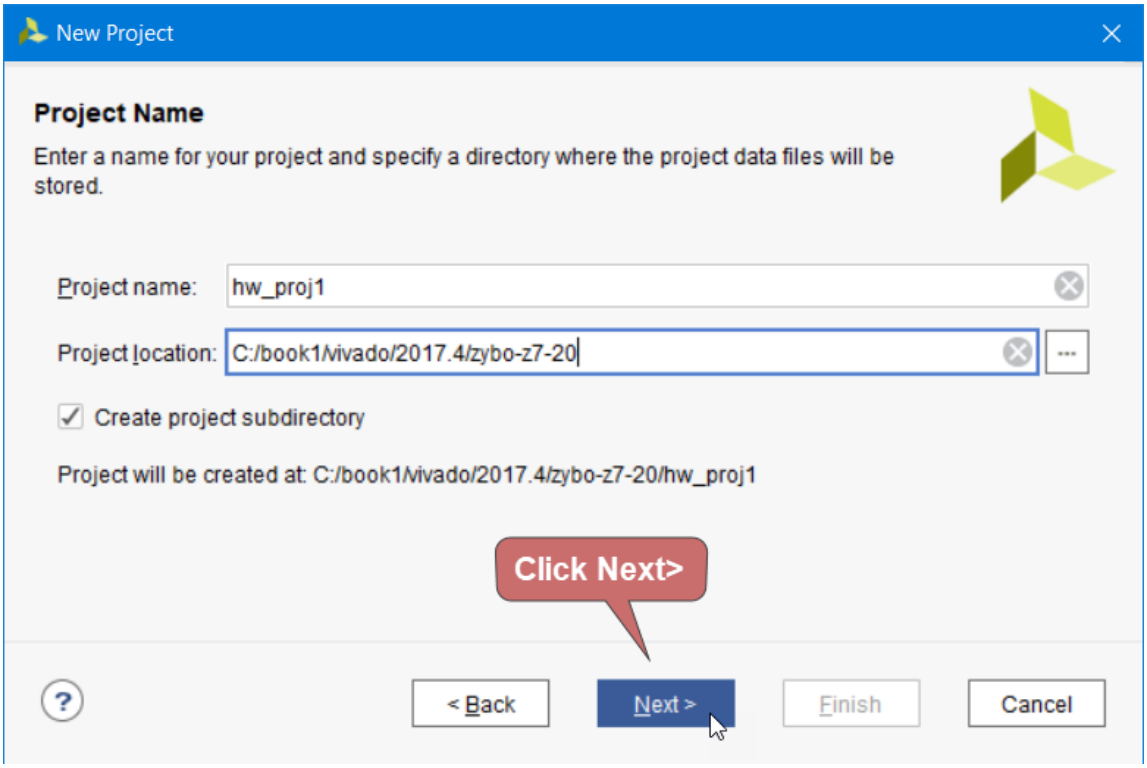


Figure 3. Choose a suitable project location and name ("hw_proj1" is used here).

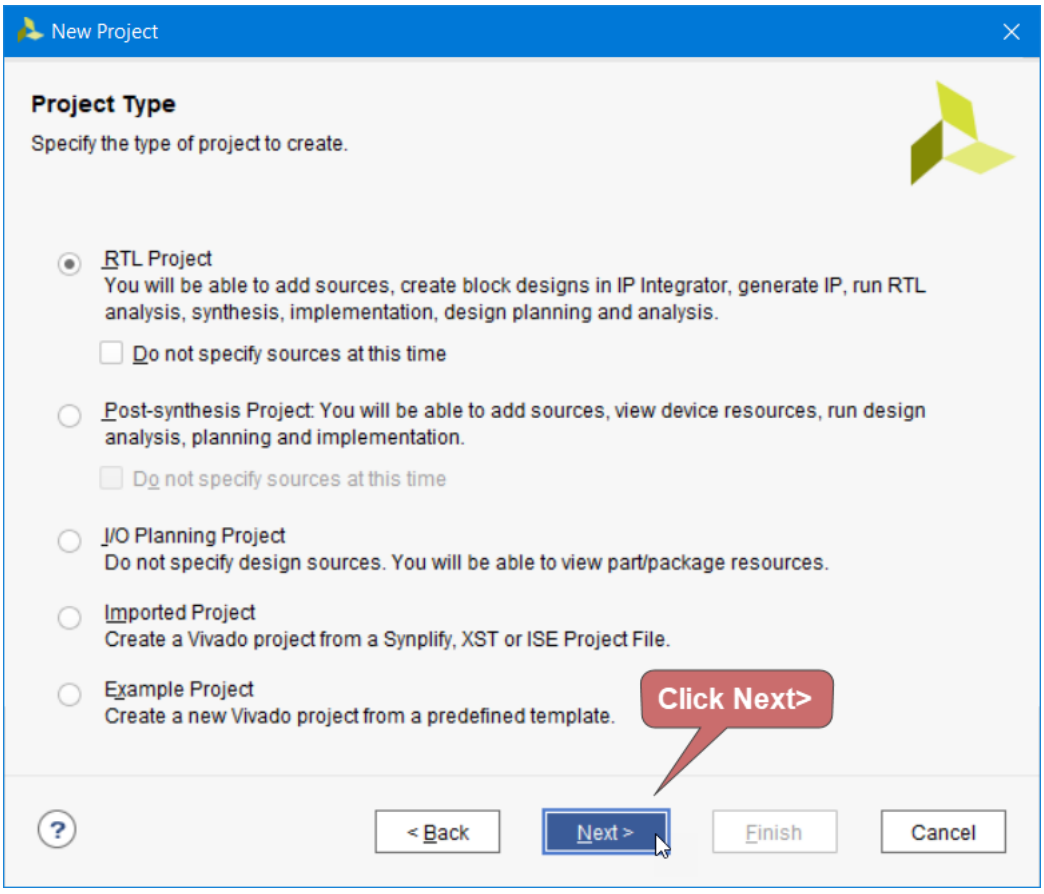


Figure 4. Keep the defaults (RTL Project) and click Next to continue.

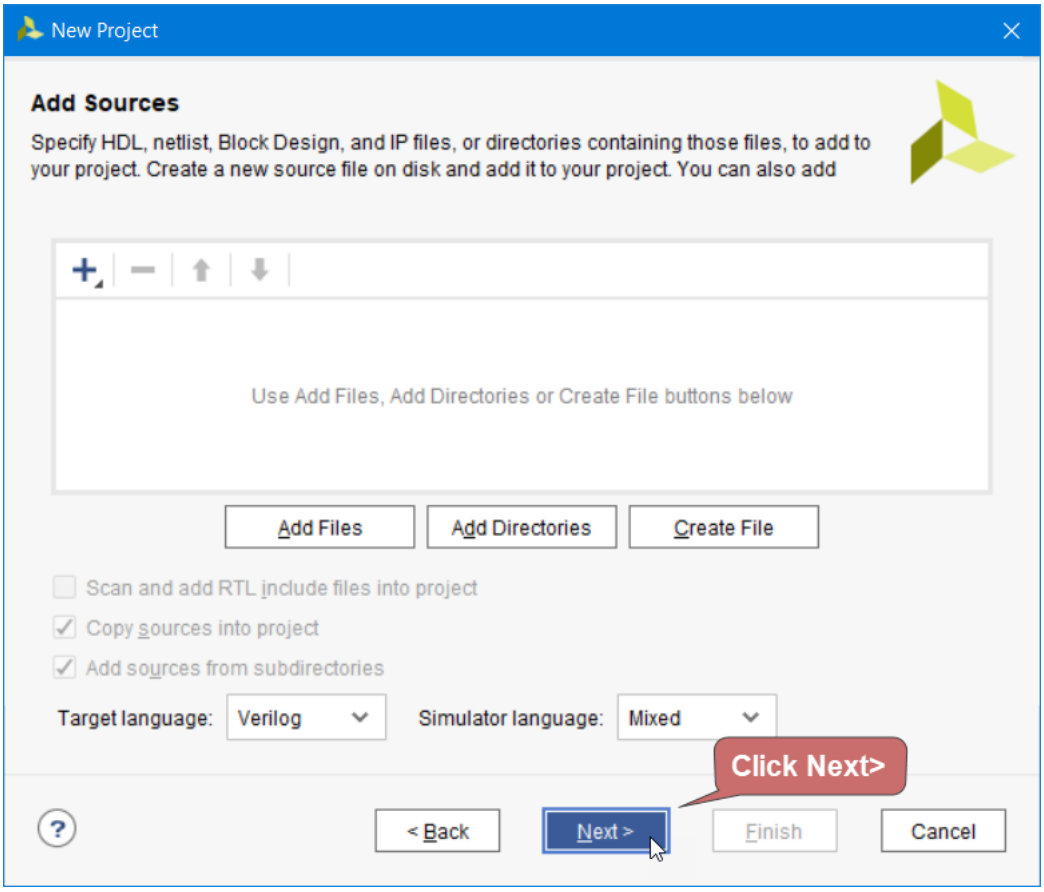


Figure 5. There are no files to add; click Next to continue.

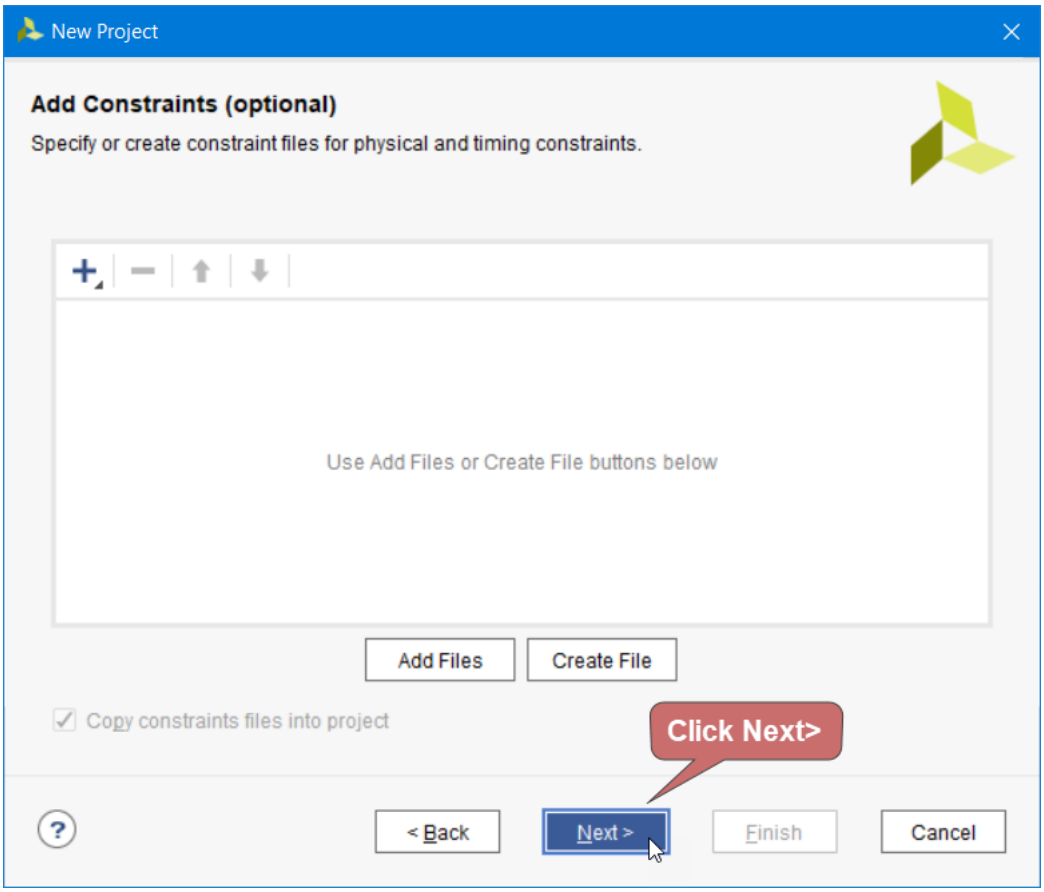


Figure 6. There are no constraints to add; click Next to continue.

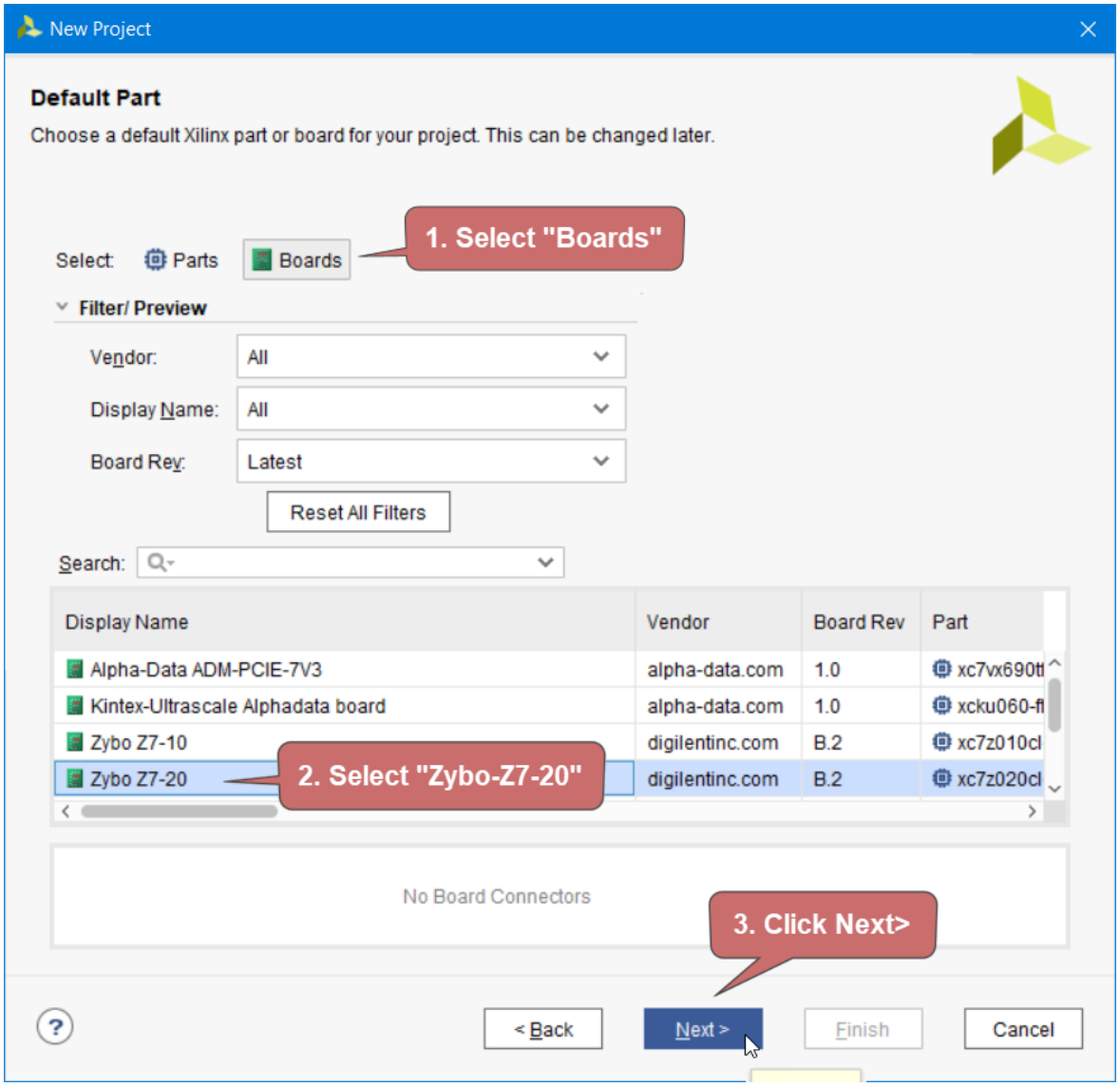


Figure 7. Choose the Digilent Zybo-Z7-20

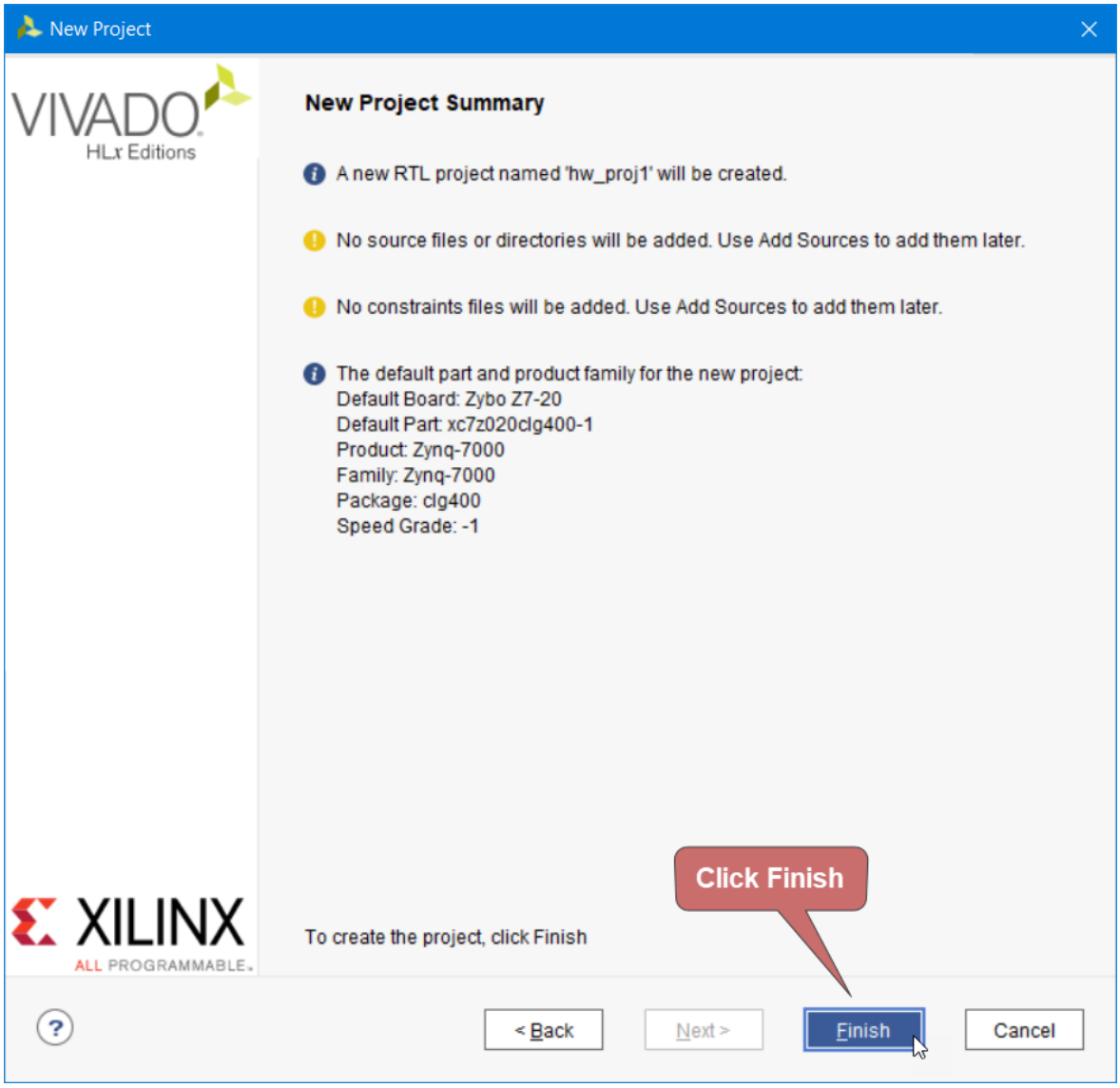


Figure 8. Click Finish.

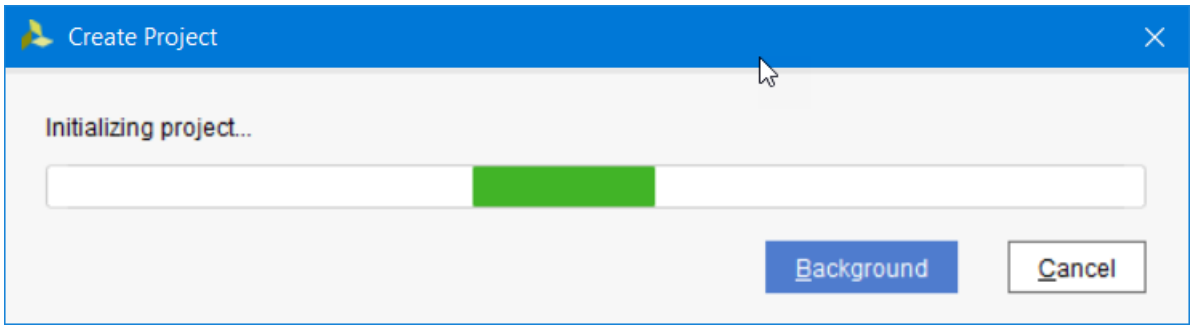


Figure 9. The project will be created...

2.2 Design Entry using IP Integrator

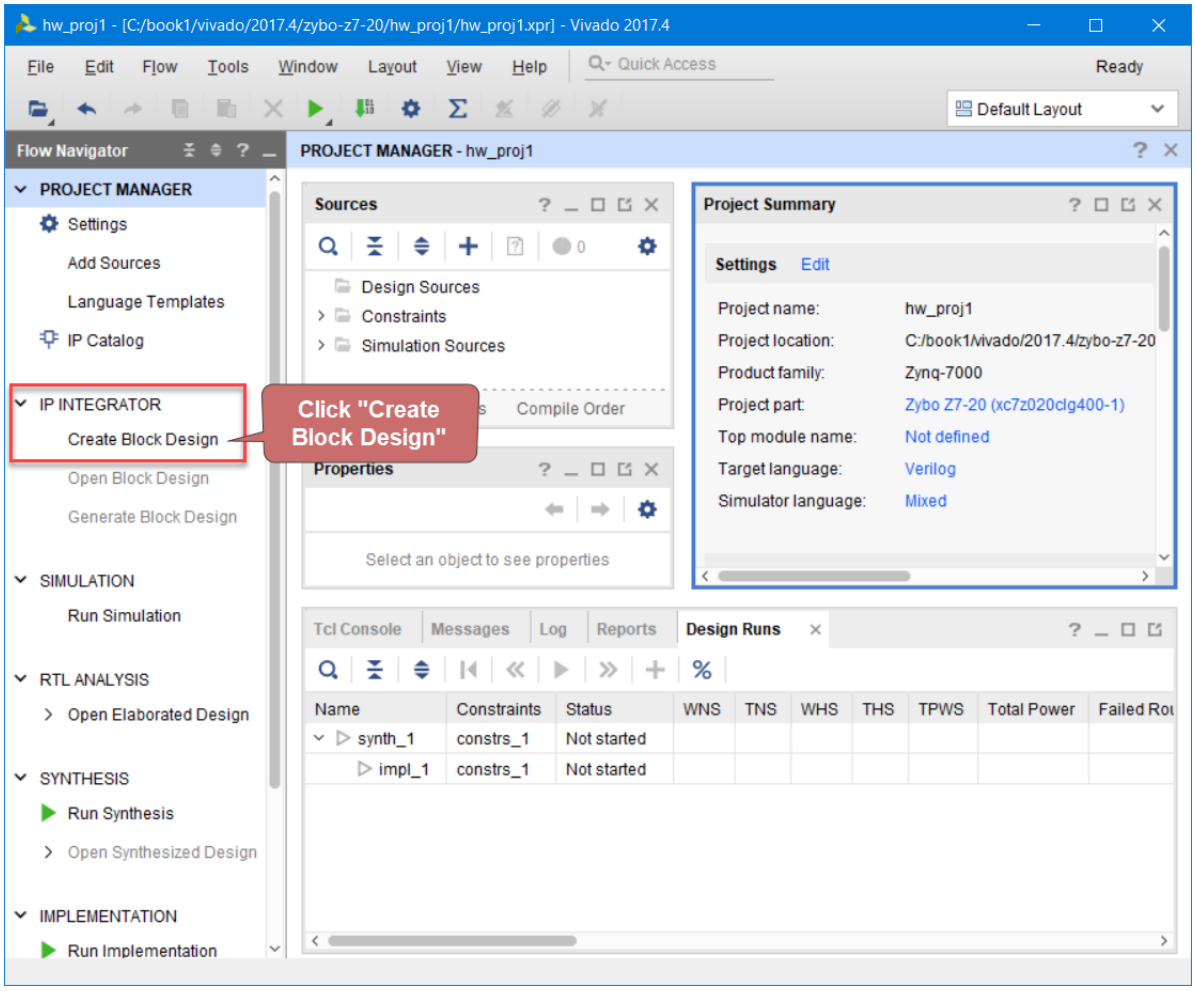


Figure 10. In the Flow Navigator, create the block design canvas.

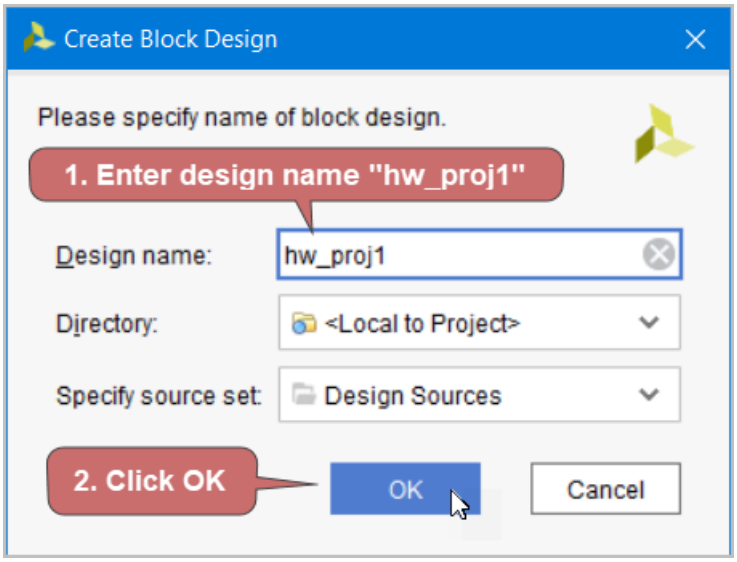


Figure 11. Enter the block design name.

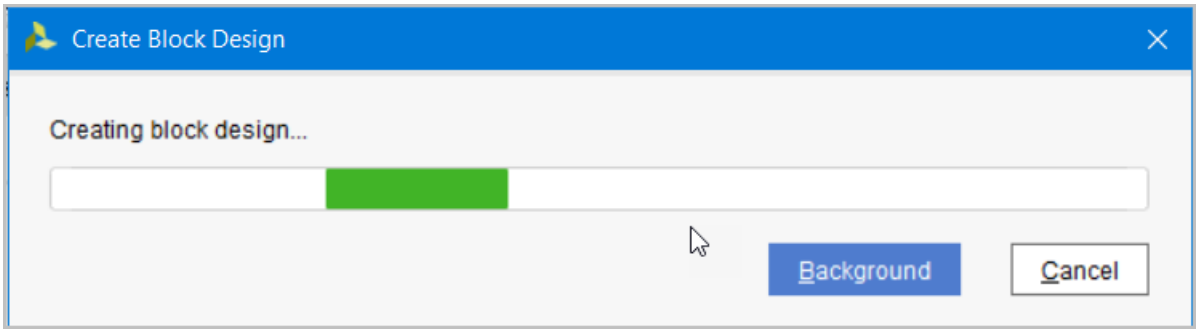


Figure 12. The block design will be created.

2.2.1 Add Zynq-7000 Processing System IP

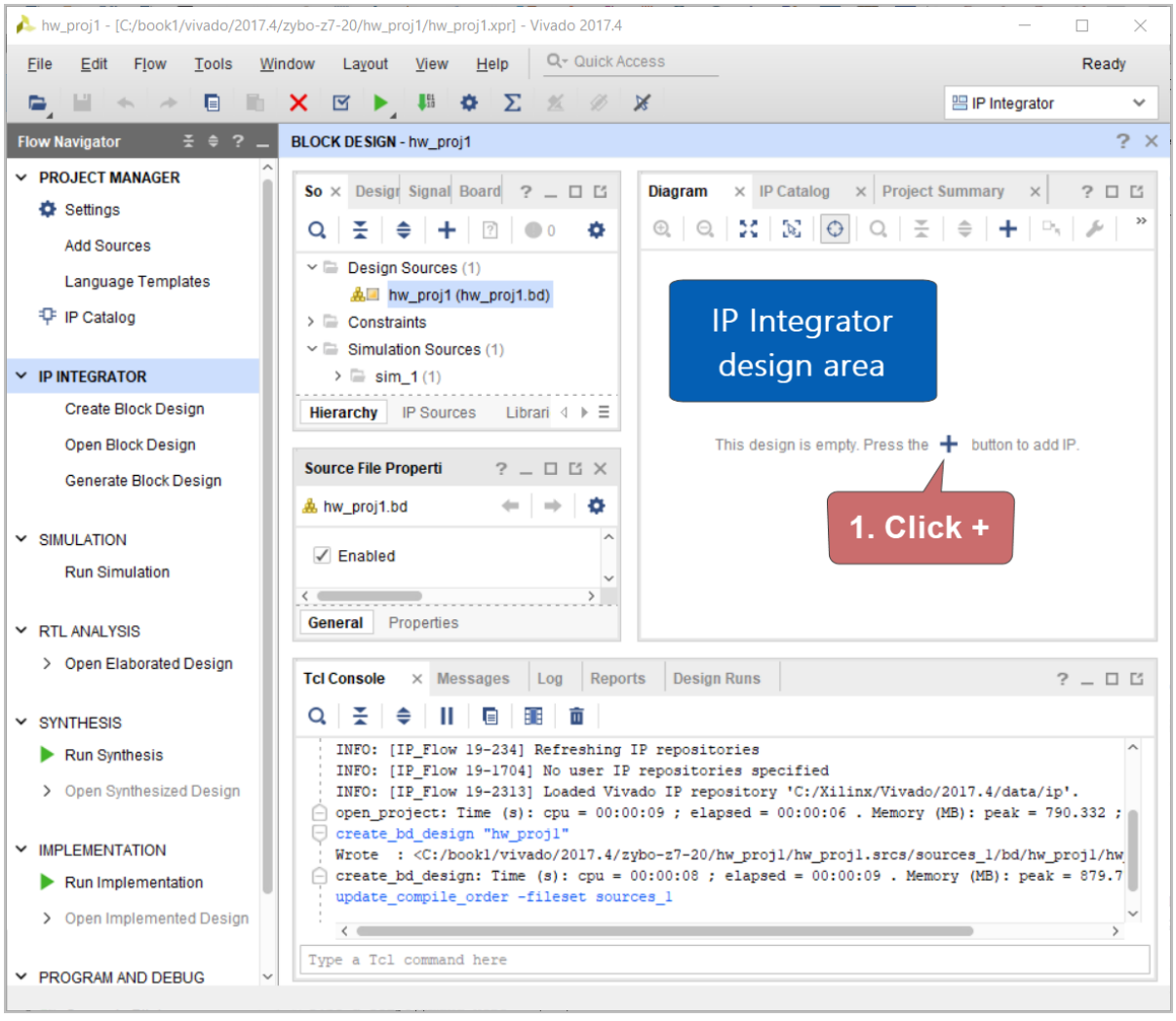


Figure 13. In the design area, click on the "+" icon, or select "CTRL + I"

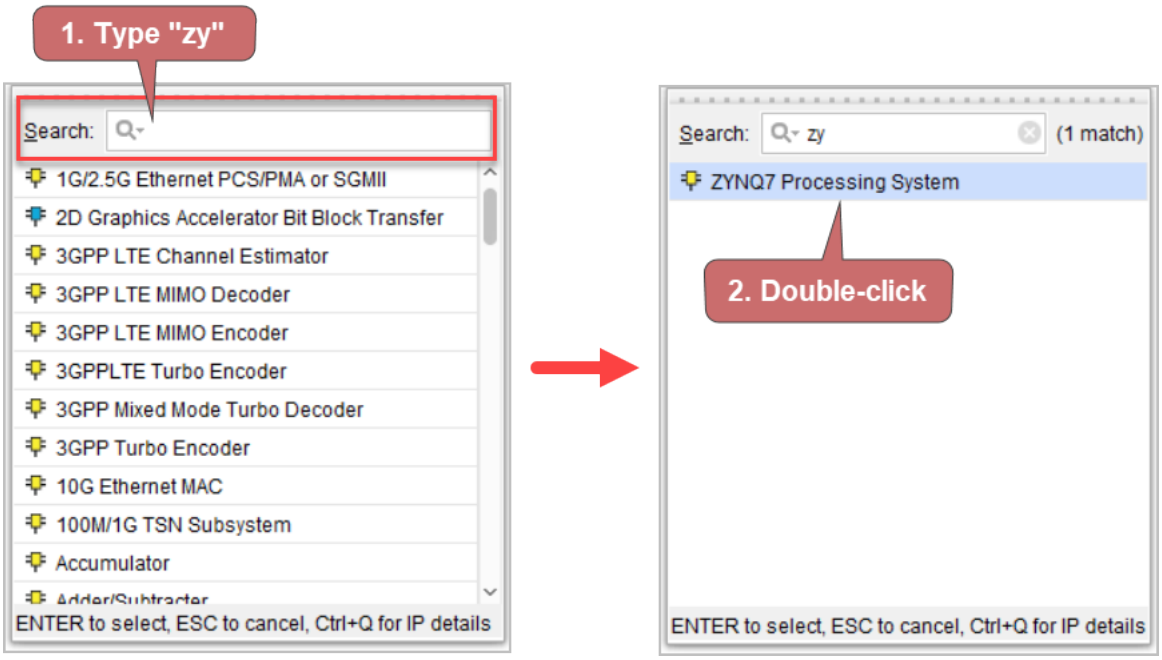


Figure 14. Find the Zynq-7000 Processing System IP

2.2.2 Generate Default Board Connections

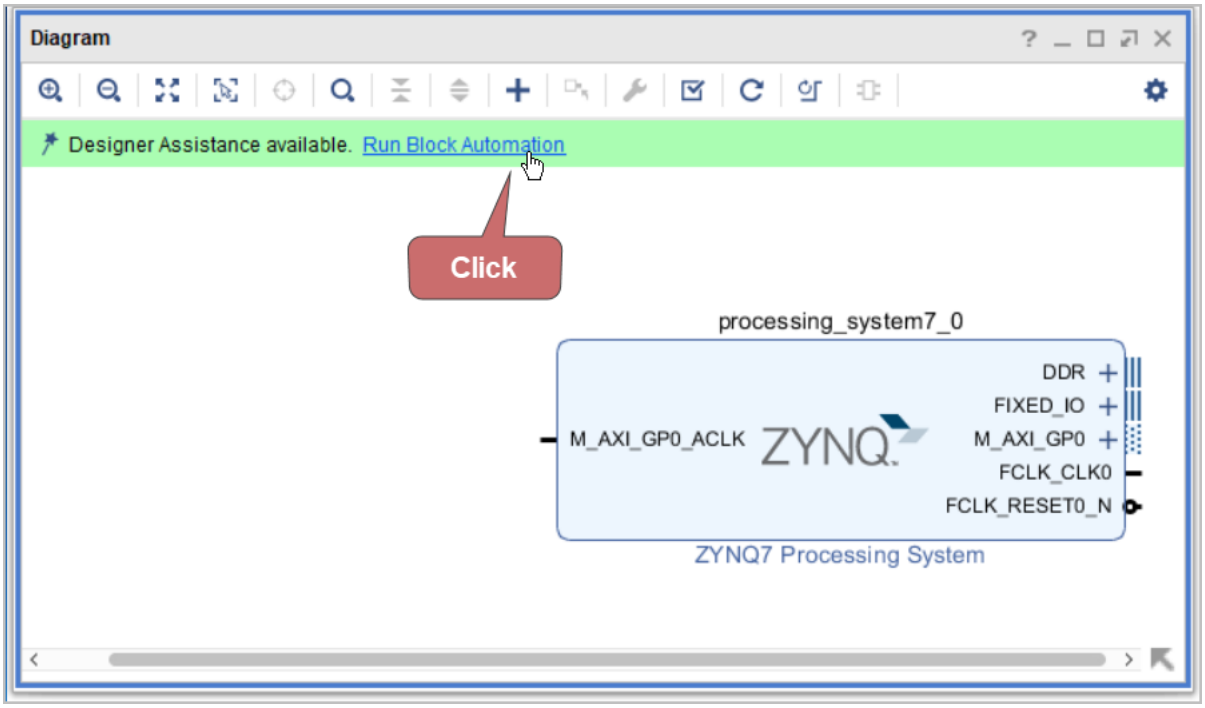


Figure 15. Click on the Run Block Automation option

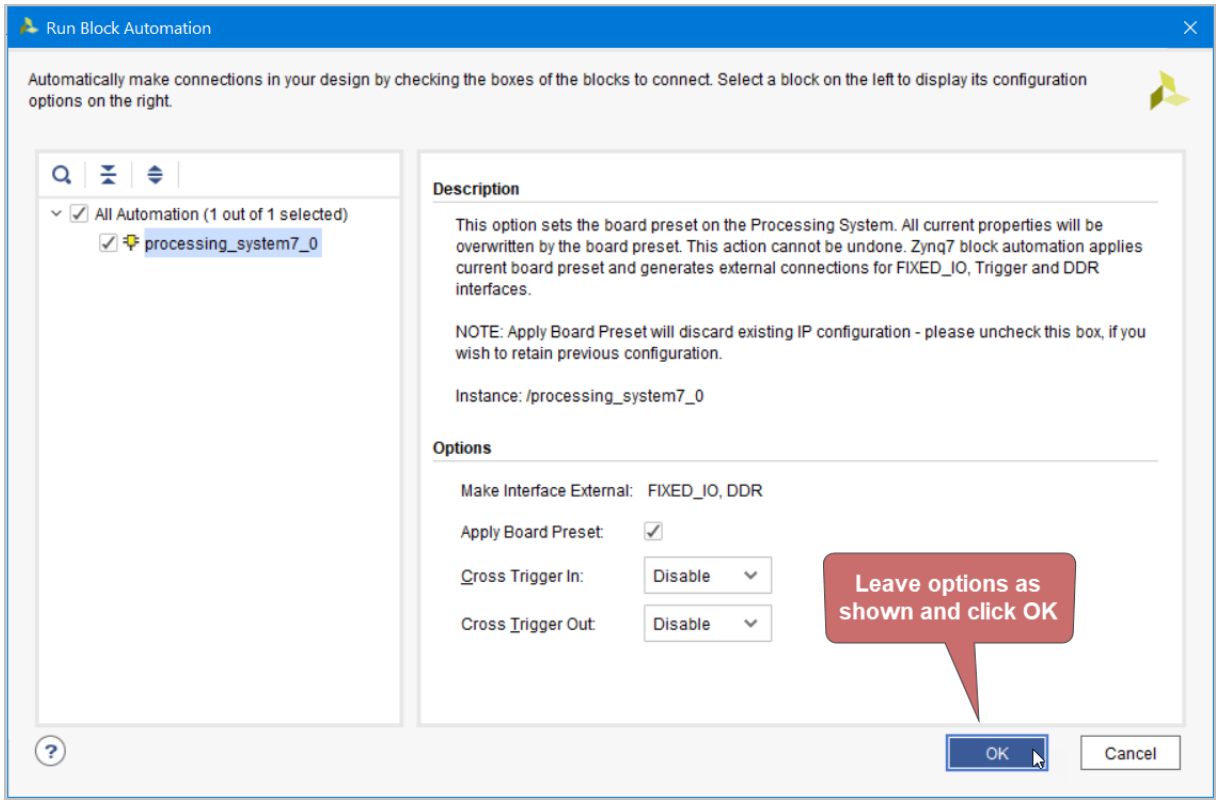


Figure 16. Select the default options and click OK.

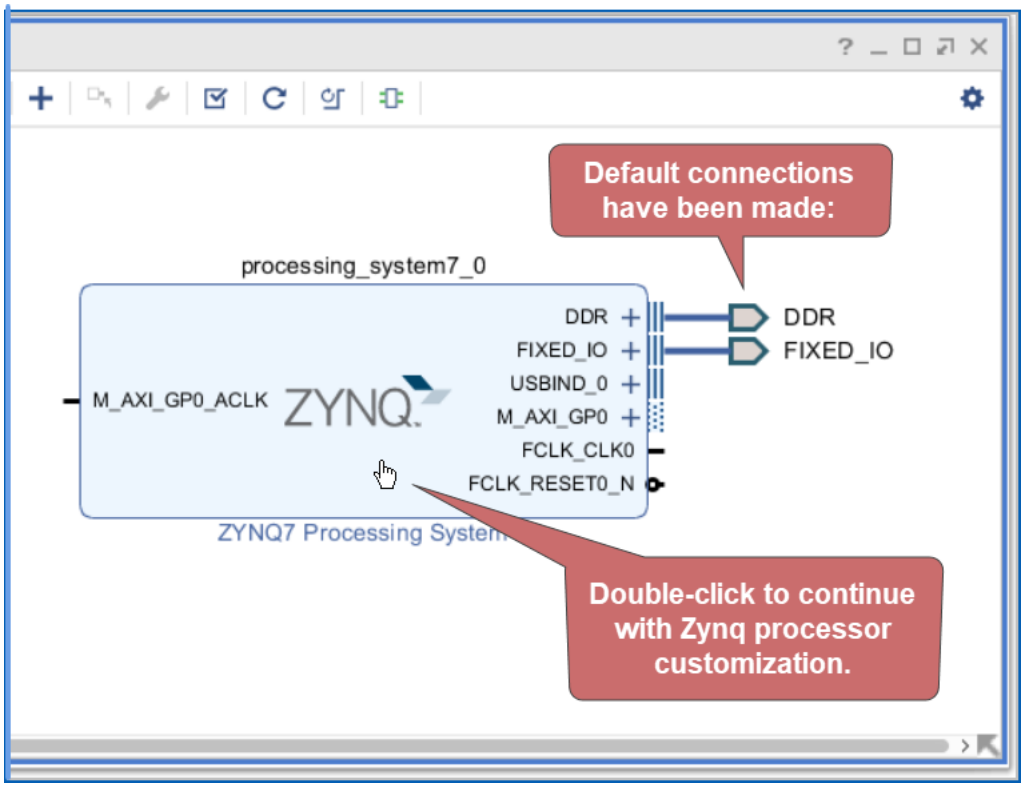


Figure 17. Double-click the Zynq IP to add more configuration options.

2.2.3 Customise the Processing System for our Project

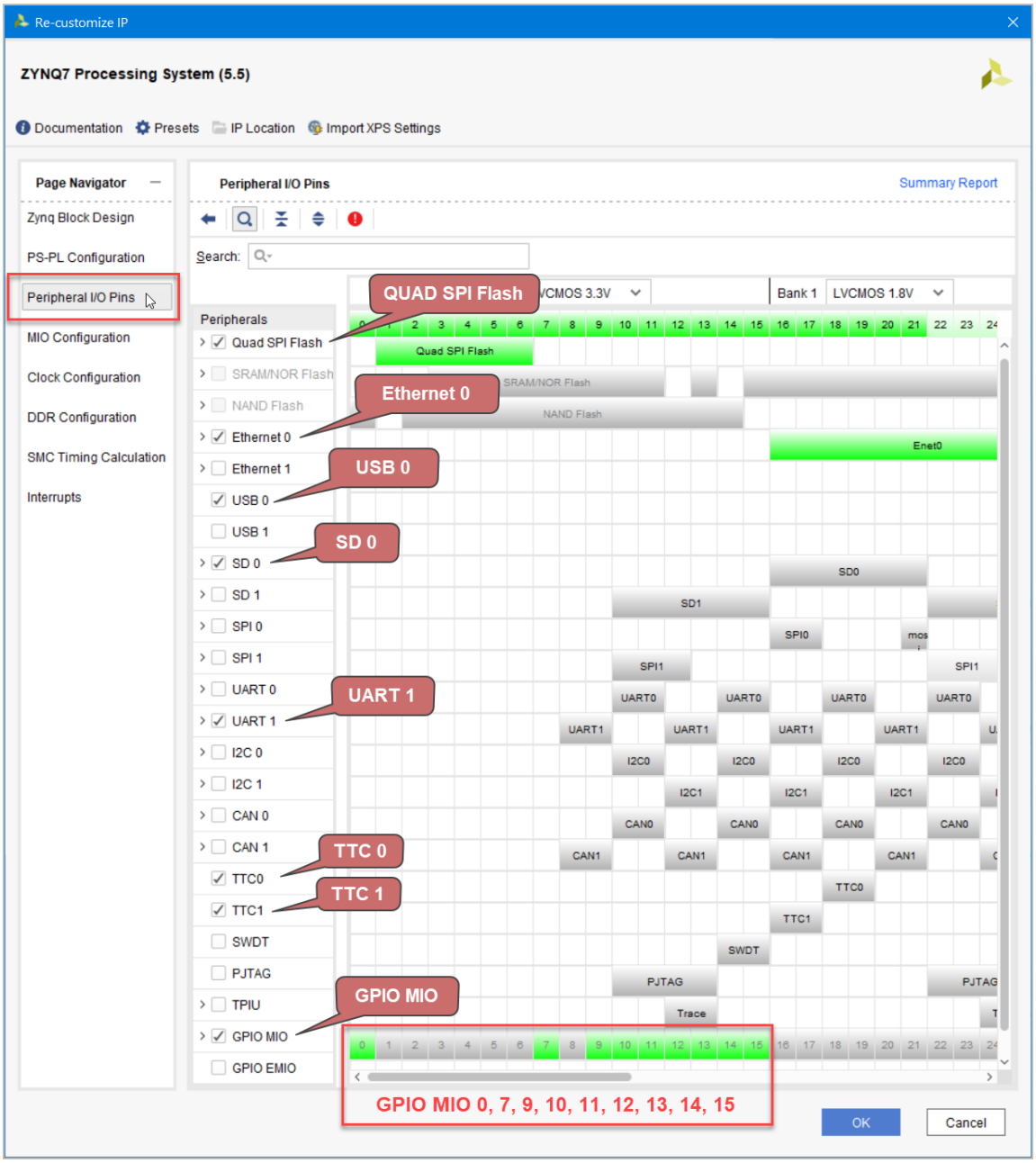


Figure 18. Peripheral I/O Pins (1). The settings should be as shown.

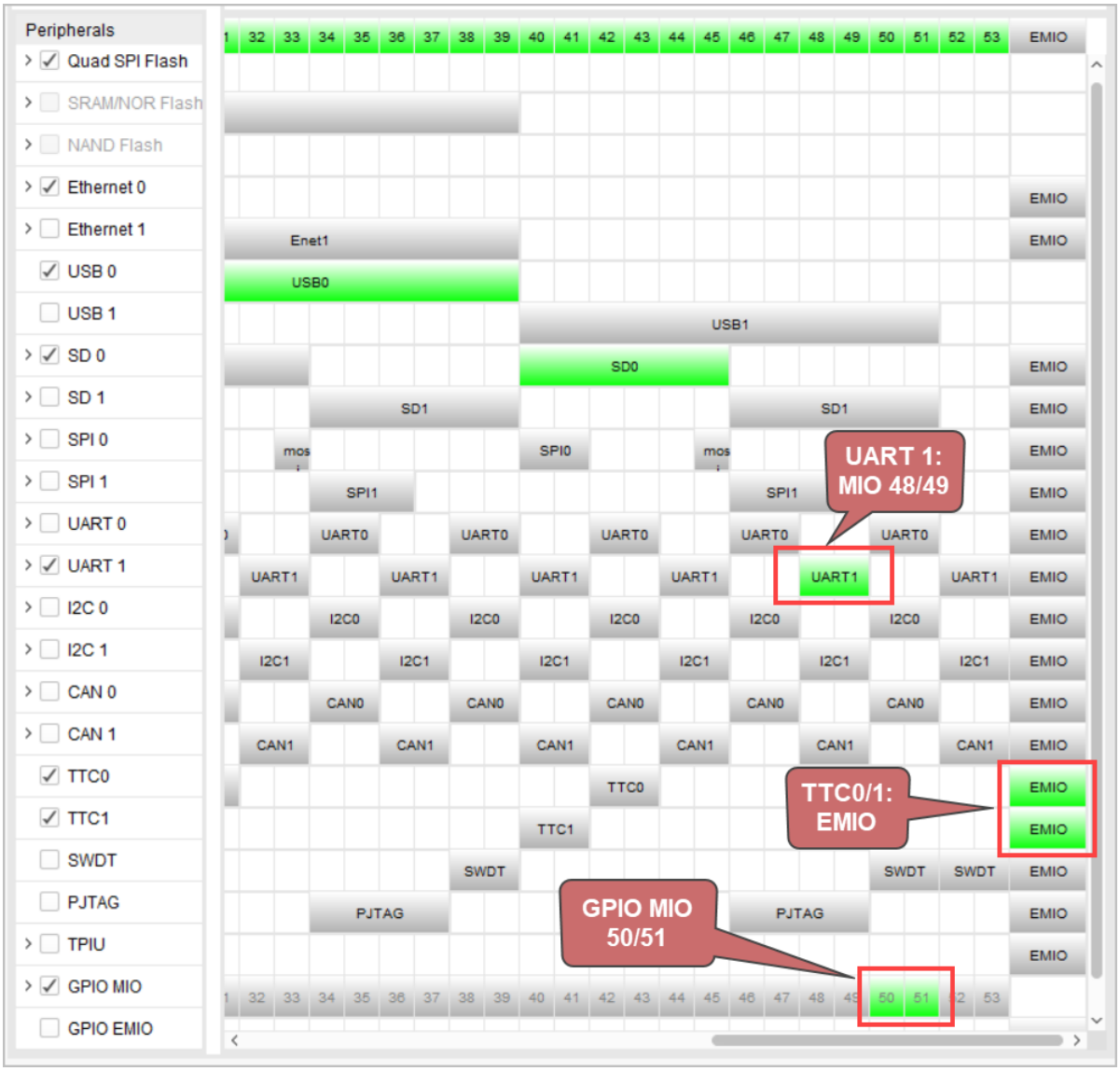


Figure 19. Enable TTC0 and TTC1 (the other settings should already be as shown).

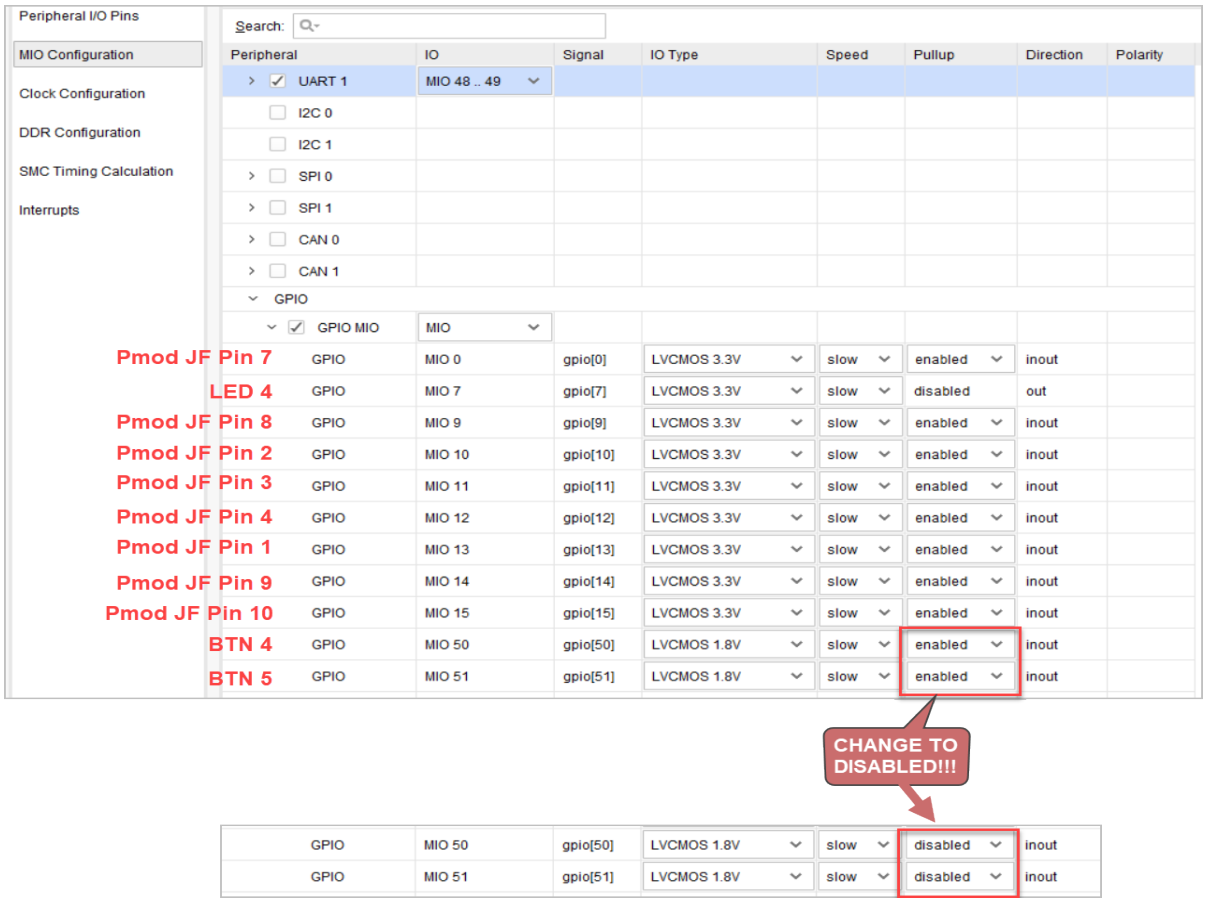


Figure 20. In the MIO Configuration Tab, the pull-ups for MIO50/MIO51 (BTN4/BTN5) can be disabled (if not already configured in the way).

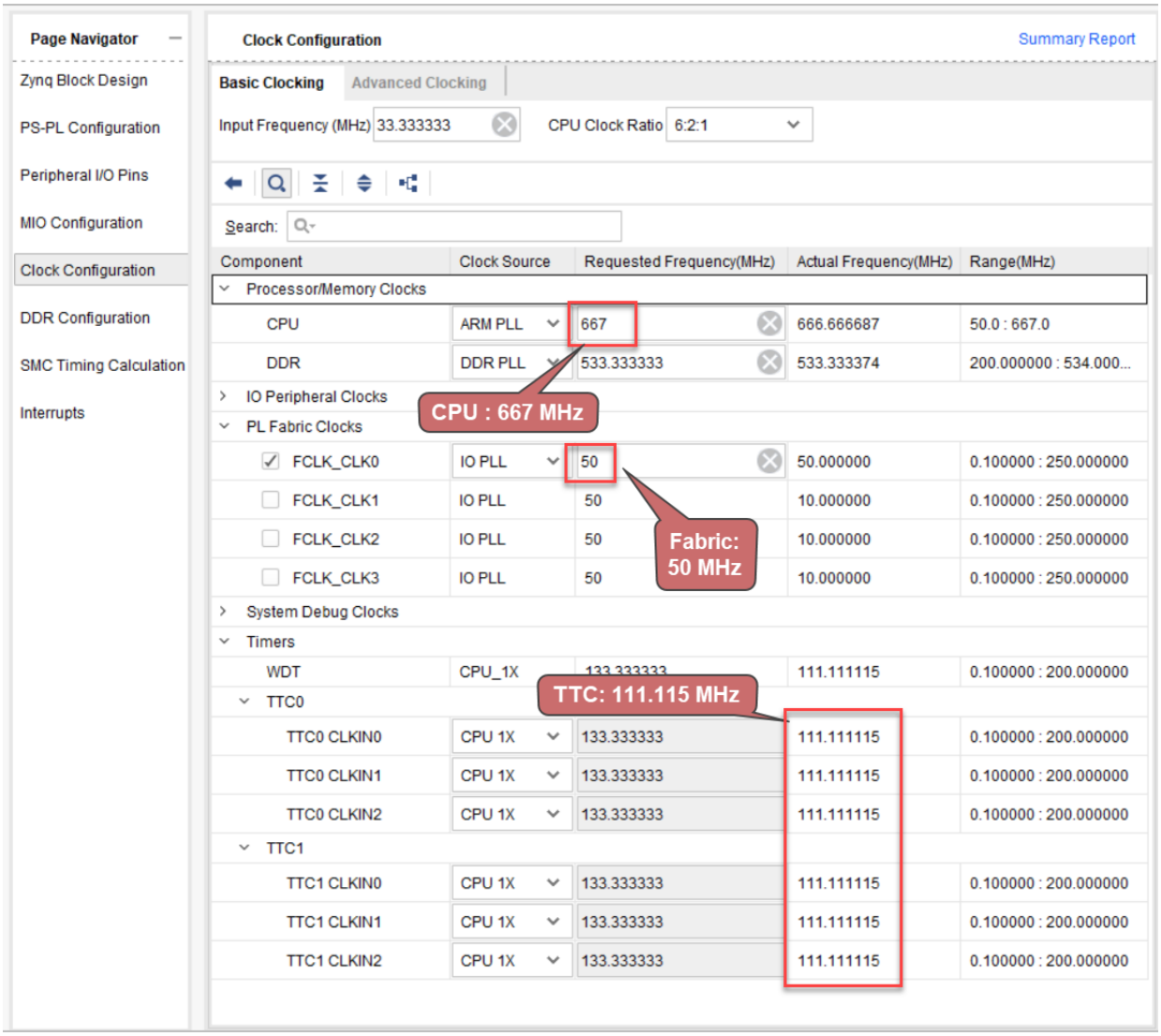


Figure 21. In the Clock Configuration Tab, verify that the clock settings are as shown.

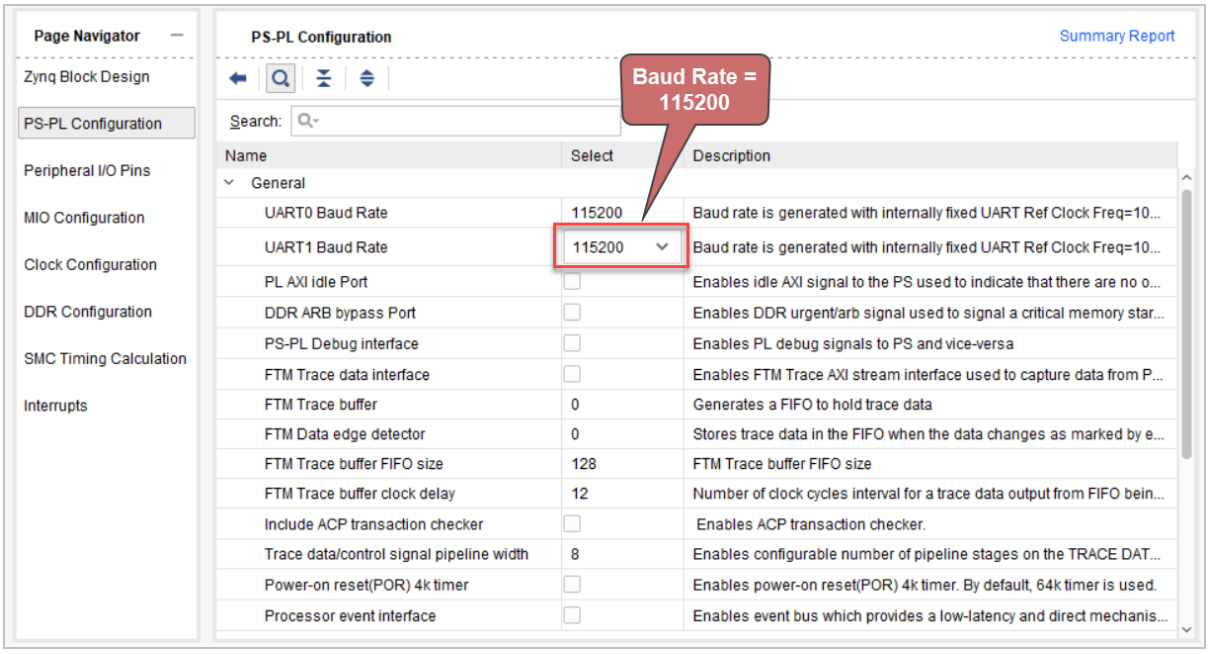


Figure 22. Check that the UART Baud Rate is 115200 baud.

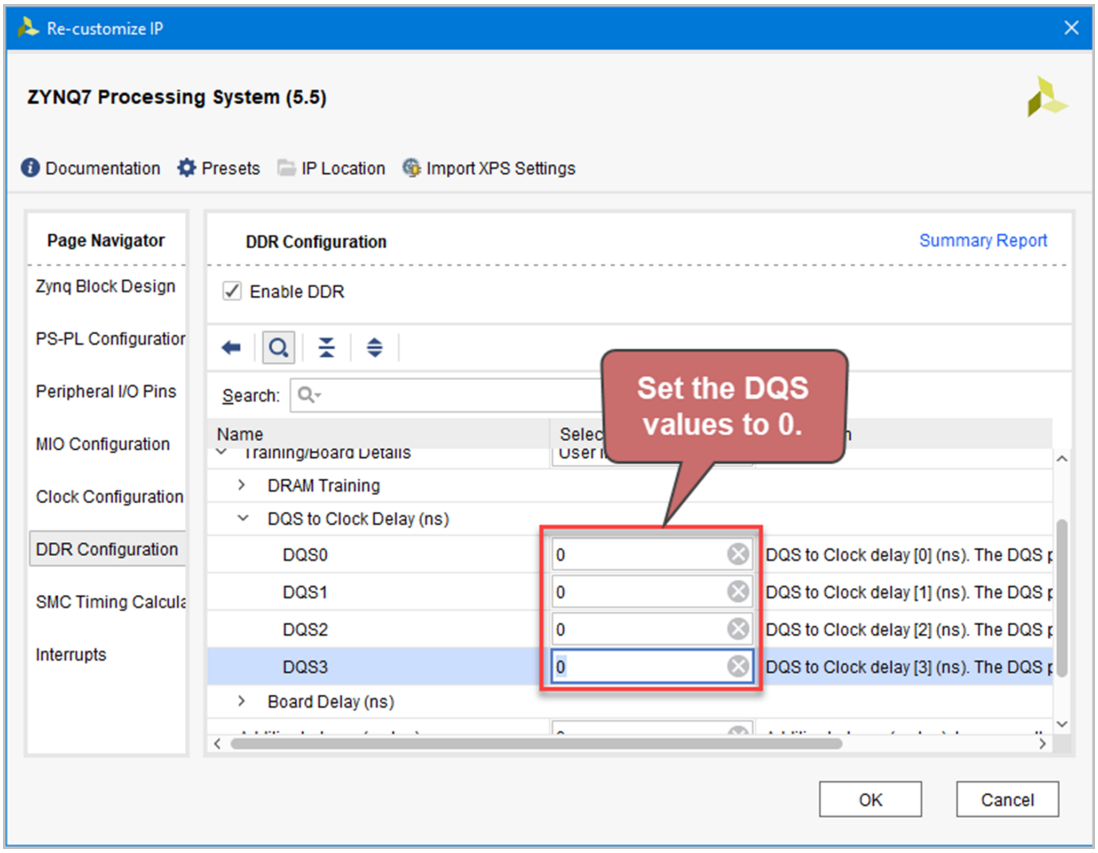


Figure 23. In the DDR Configuration tab, the DQS to Clock Delay settings can all be changed to 0.

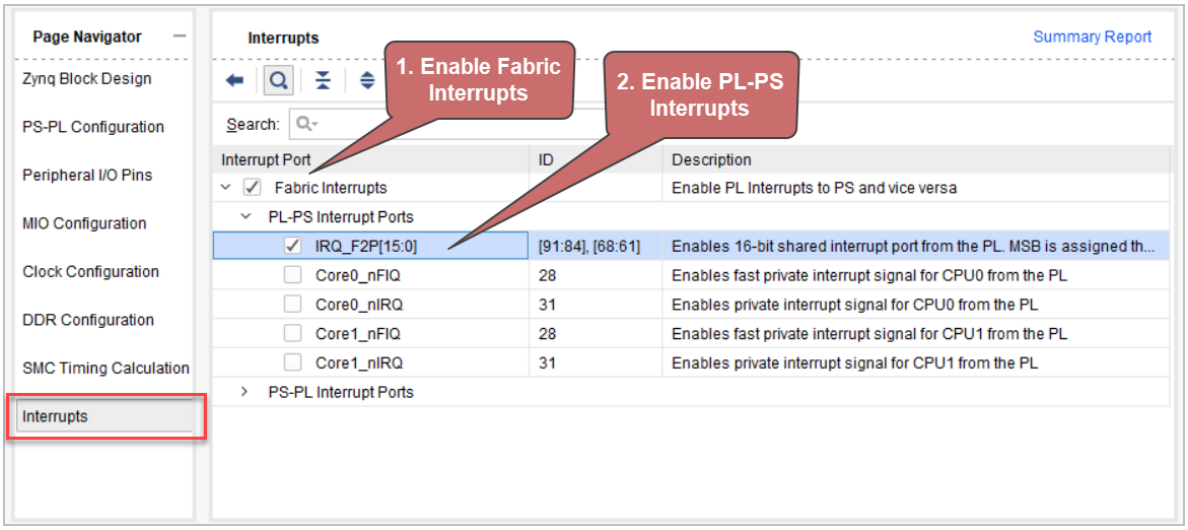


Figure 24. In the Interrupts tab, enable the IRQ_F2P[15:0] interrupts.

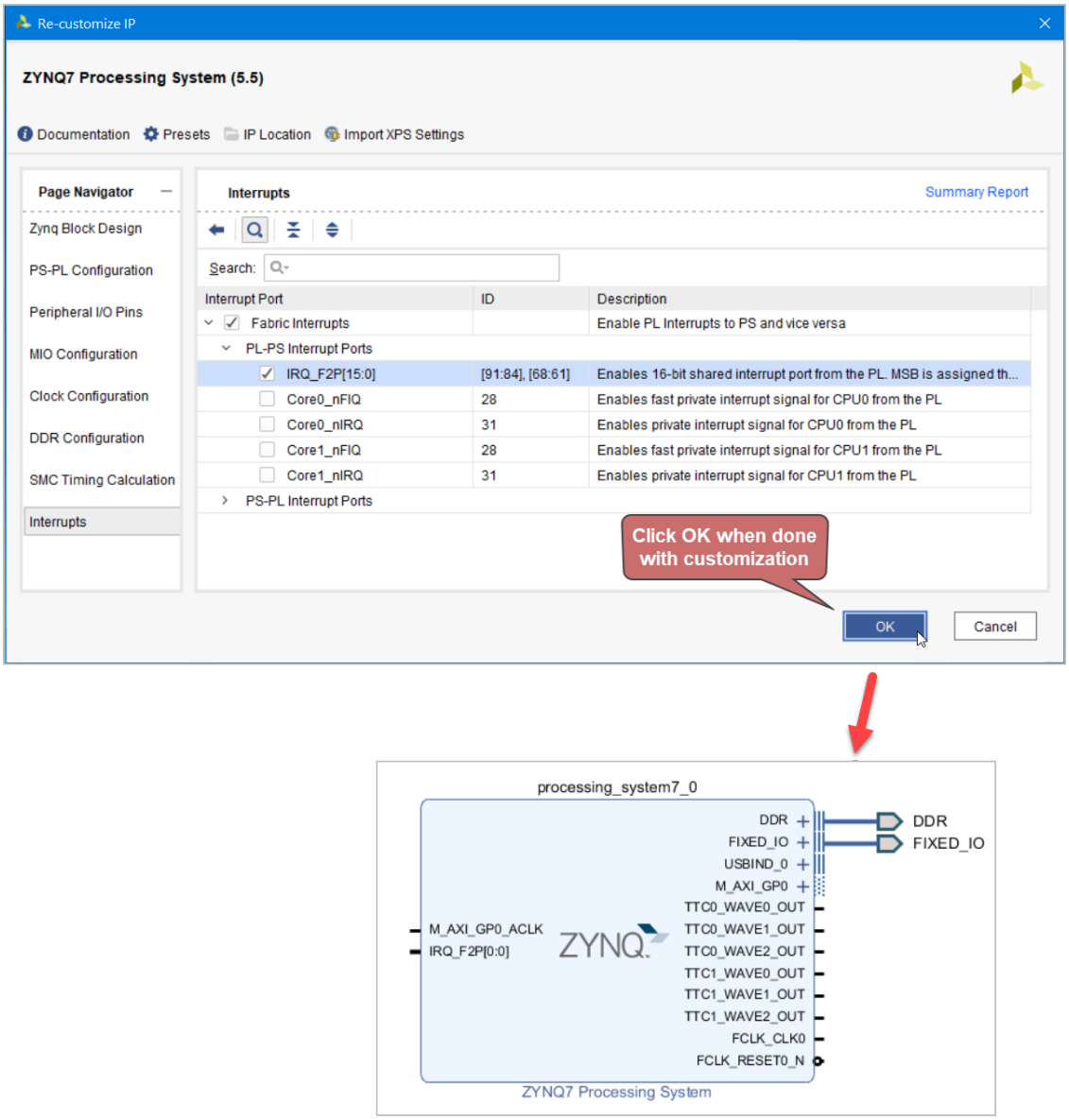


Figure 25. Click OK to apply customisation.

2.2.4 Add TTC Output Pins

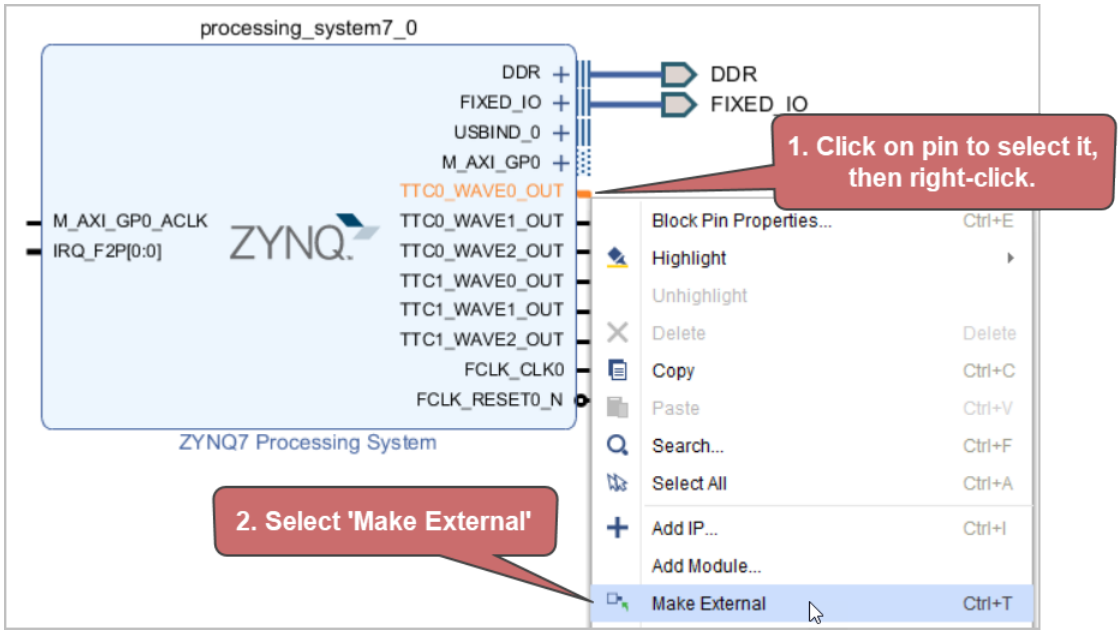


Figure 26. Add TTC pins by right-clicking and selecting "Make External"

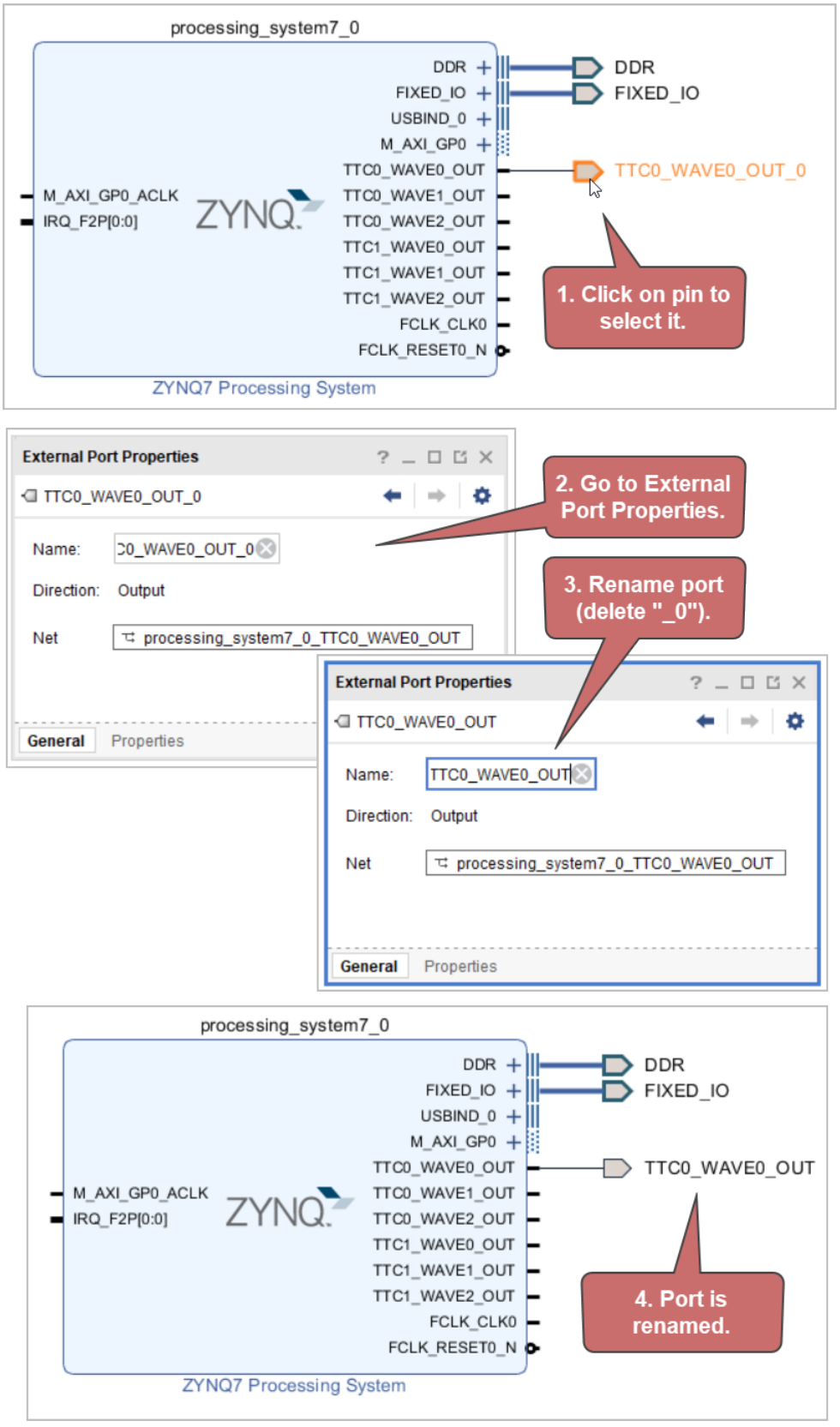


Figure 27. Rename the pins: delete "_0".

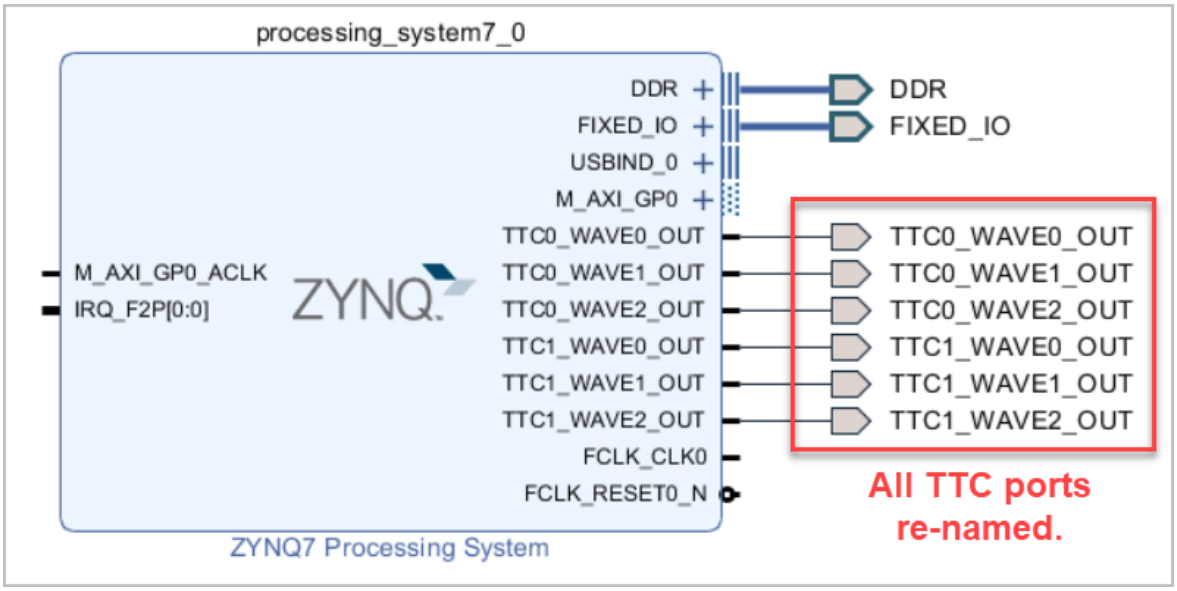


Figure 28. Repeat for all TTC pins

2.2.5 Add AXI GPIO IP

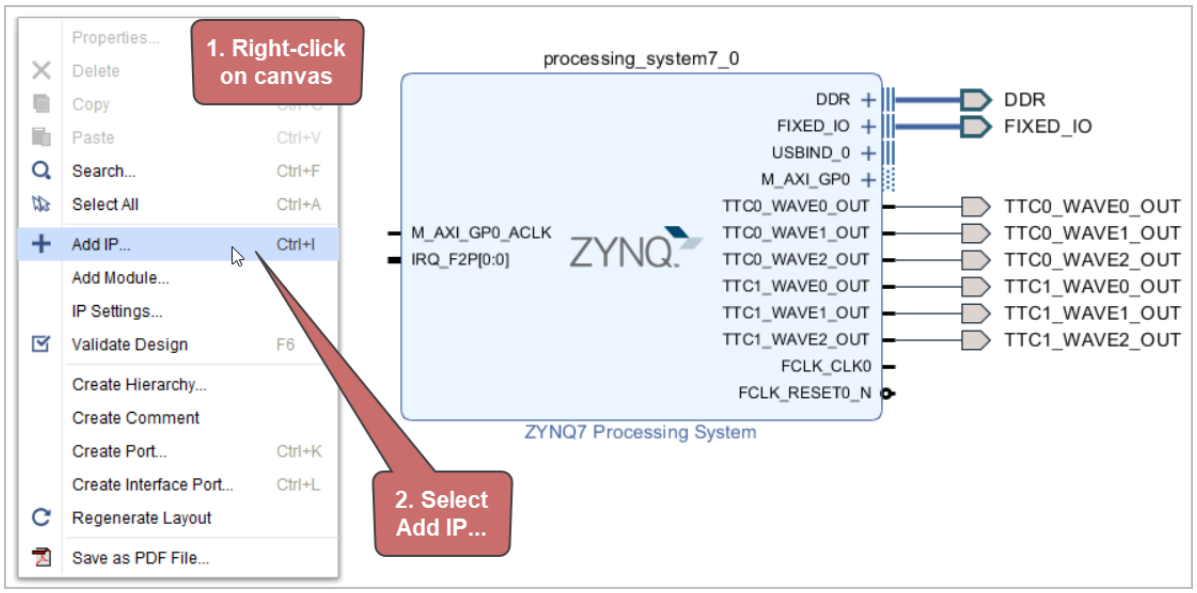


Figure 29. Add the AXI GPIO IP (1).

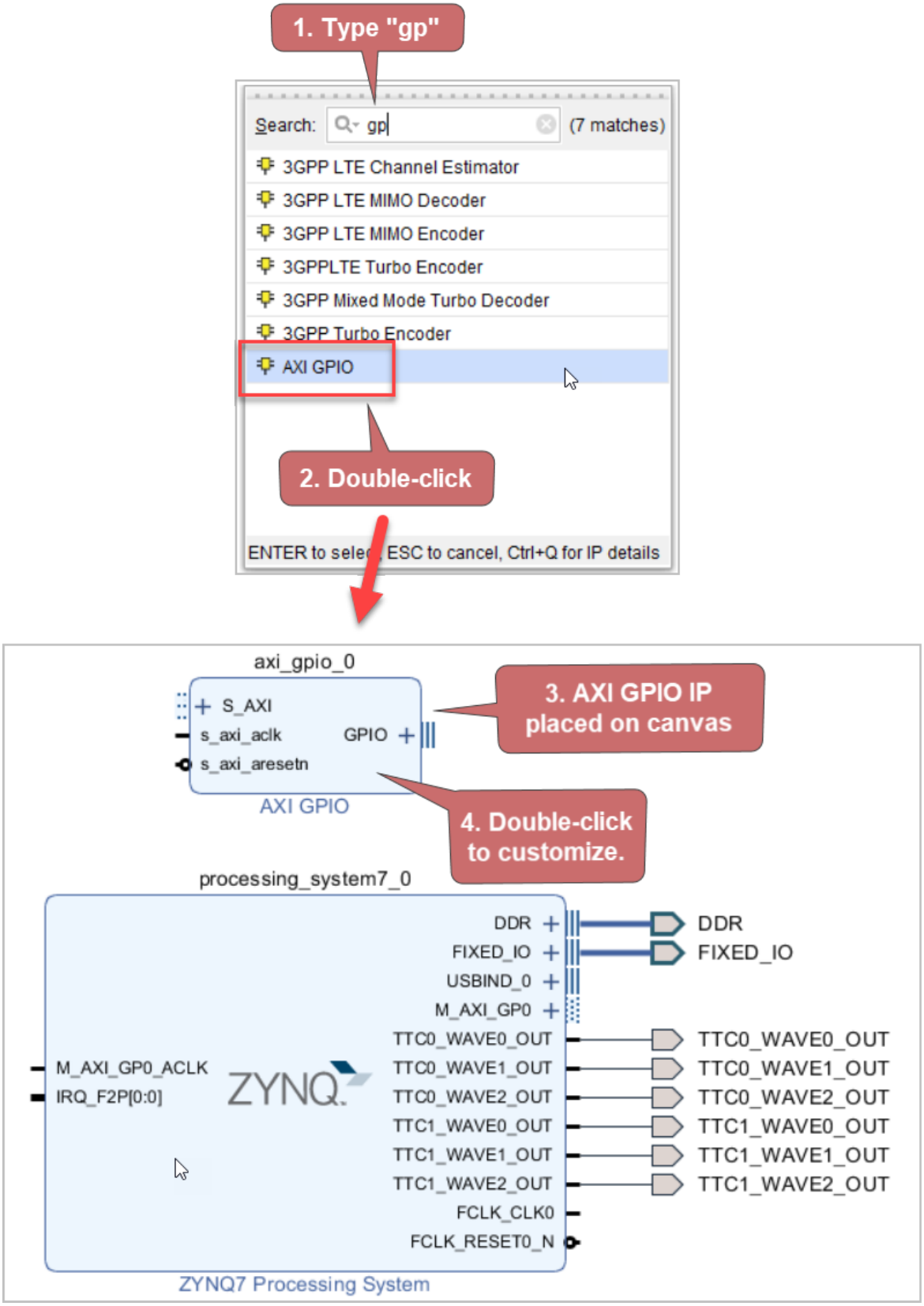


Figure 30. Add the AXI GPIO IP (2).

2.2.5.1 Customise the AXI GPIO IP

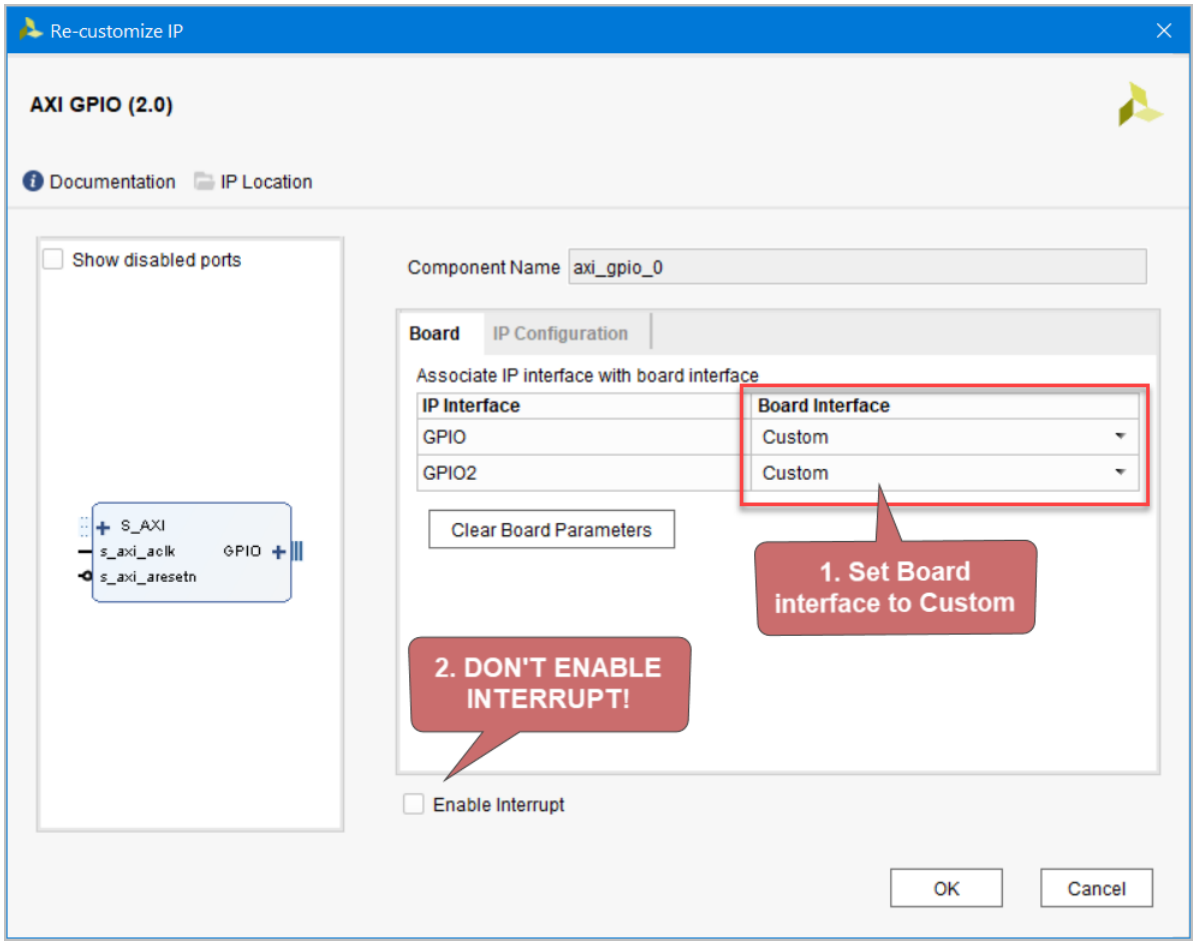


Figure 31. Set the GPIO Board options

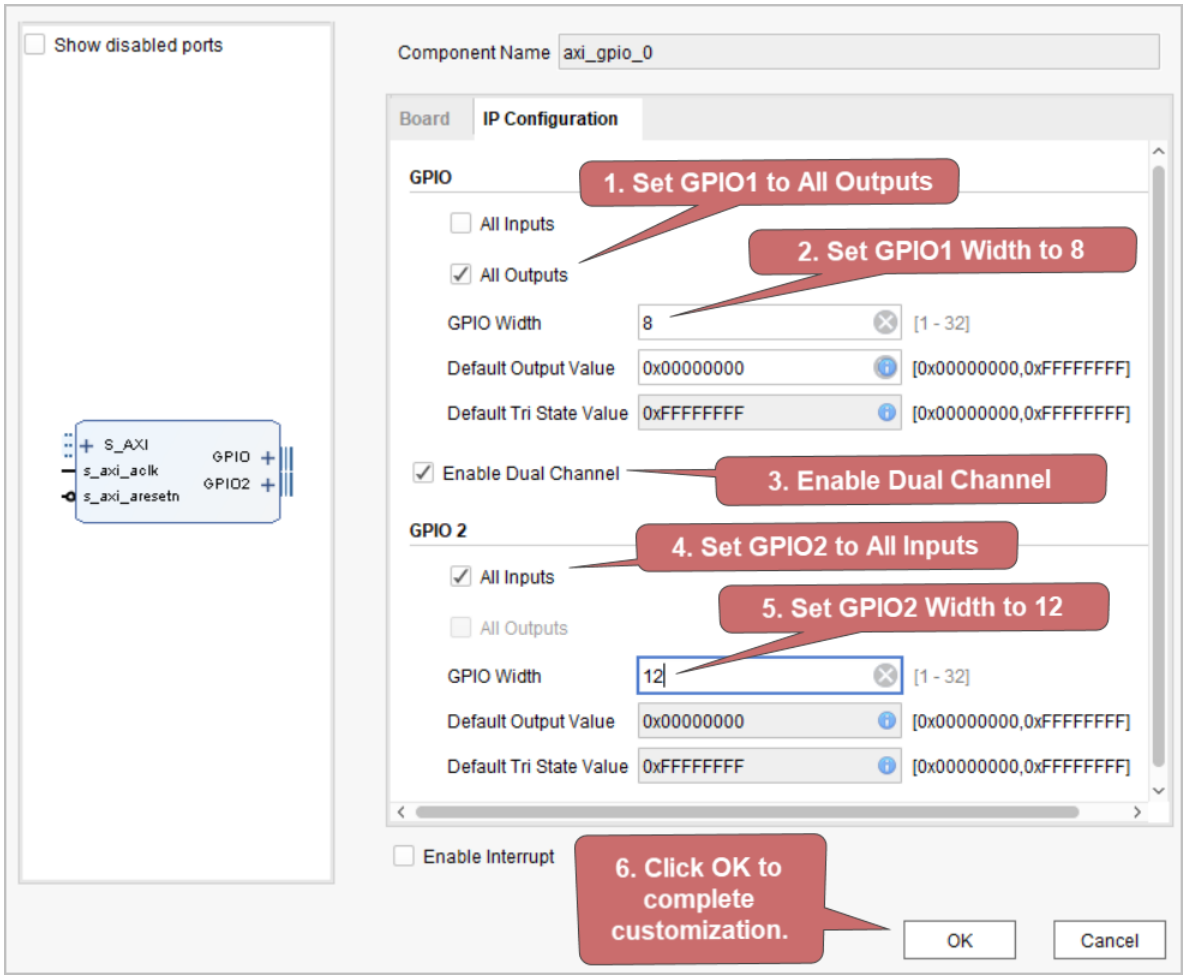


Figure 32. Set GPIO IP Configuration.

2.2.5.2 Add GPIO Input/Output Pins

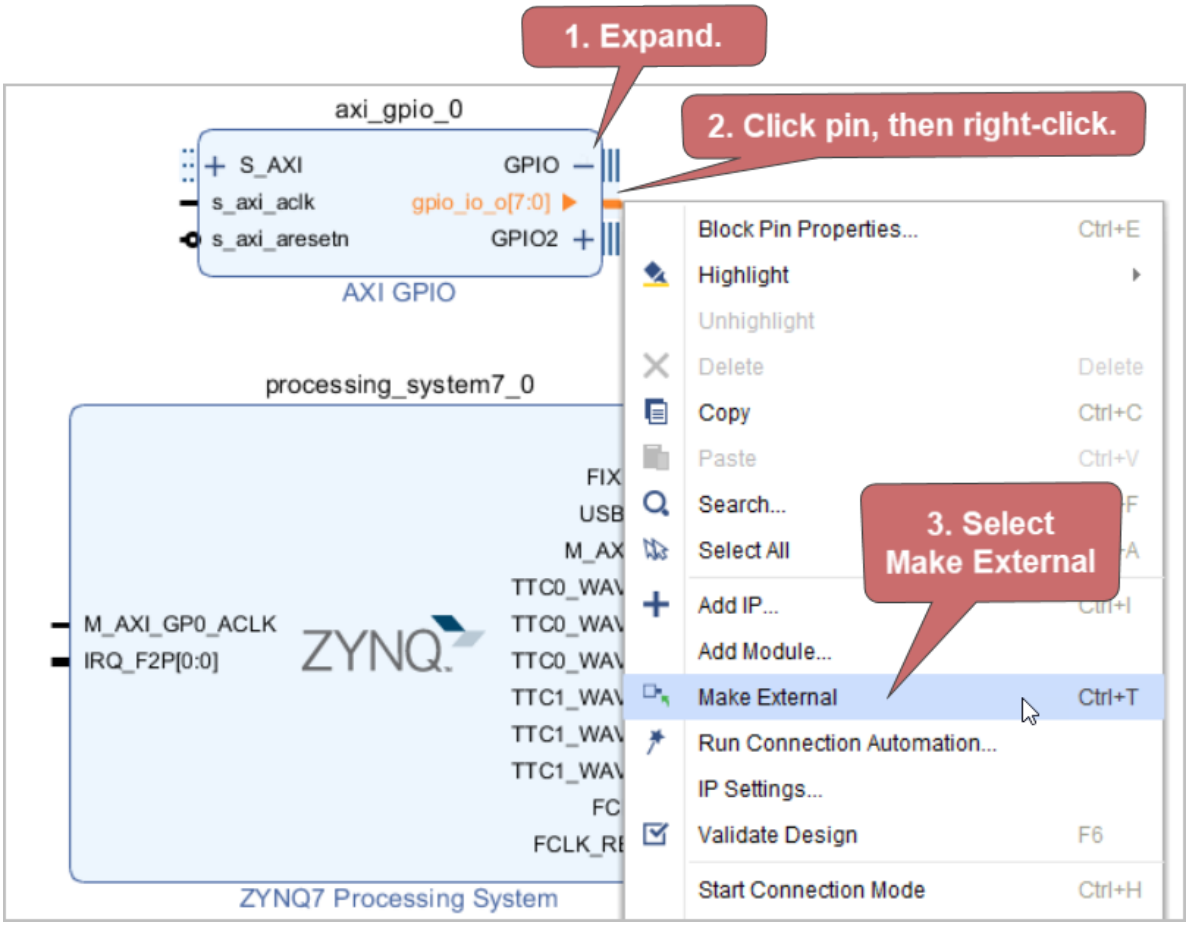


Figure 33. Add the output pins for channel 1.

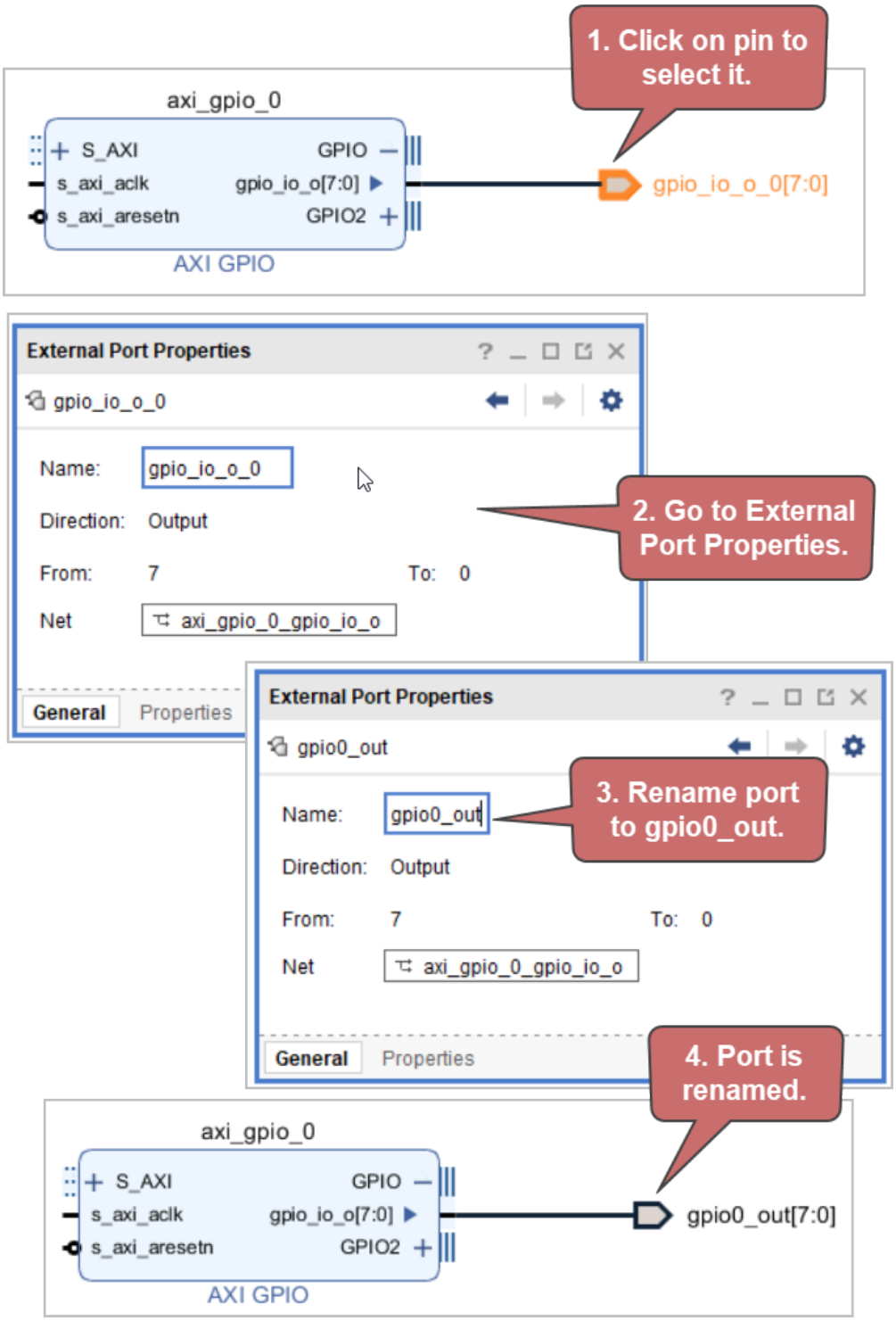


Figure 34. Rename the output pins.

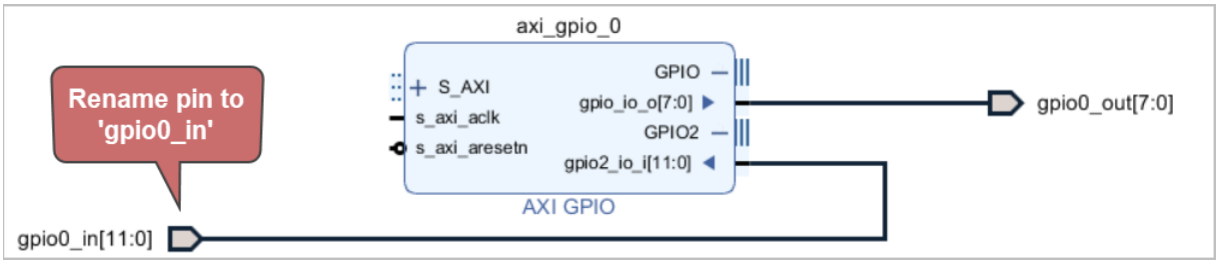


Figure 35. Repeat for the channel 2 input pins.

2.2.5.3 Run Connection Automation to Update Block Design

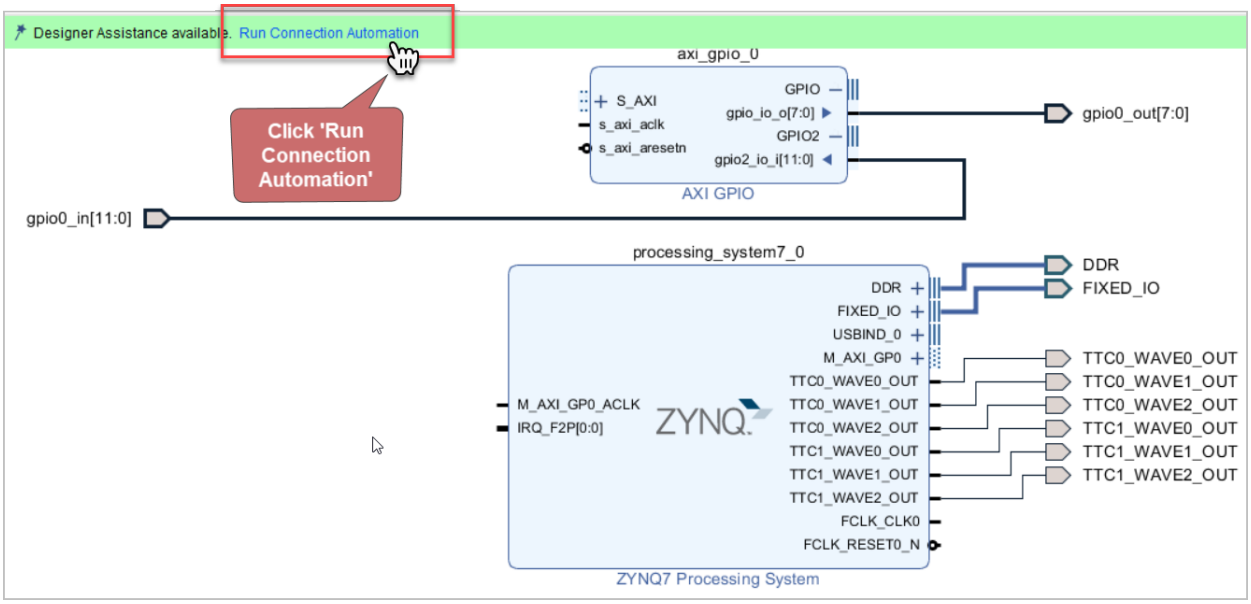


Figure 36. Run Connection Automation.

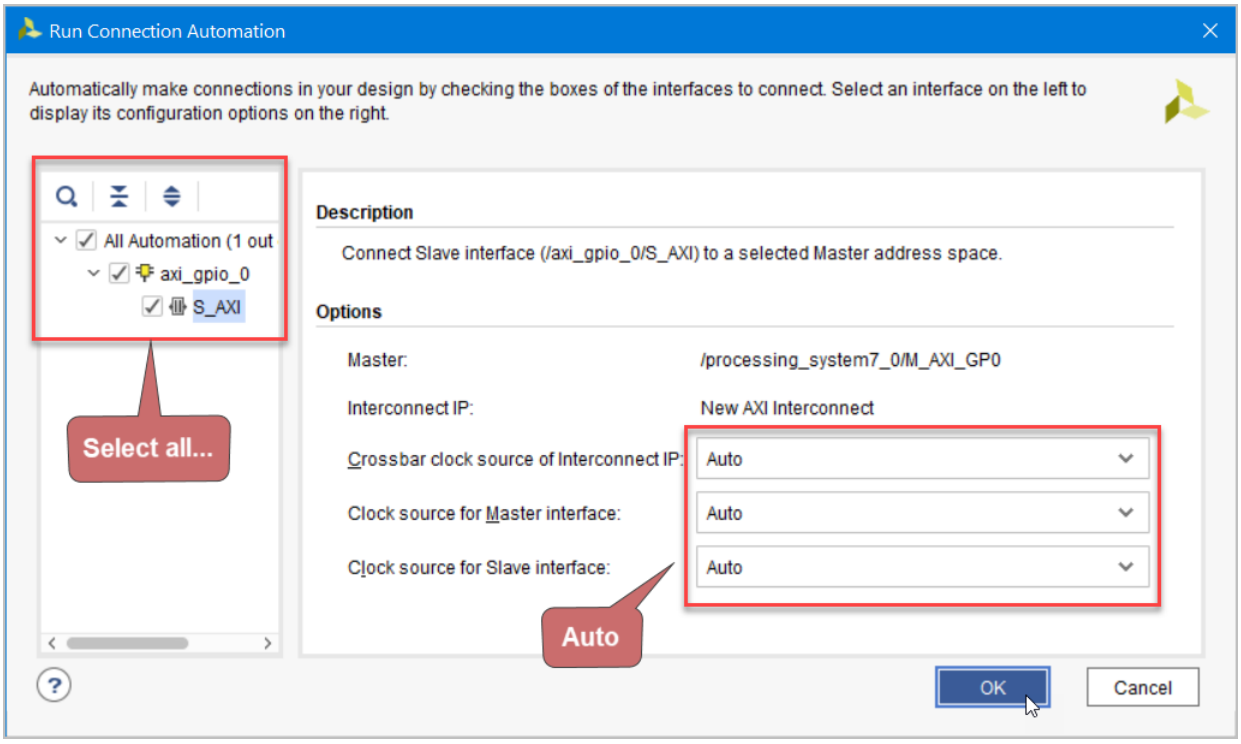


Figure 37. Ensure the options are set to Auto, and click OK.

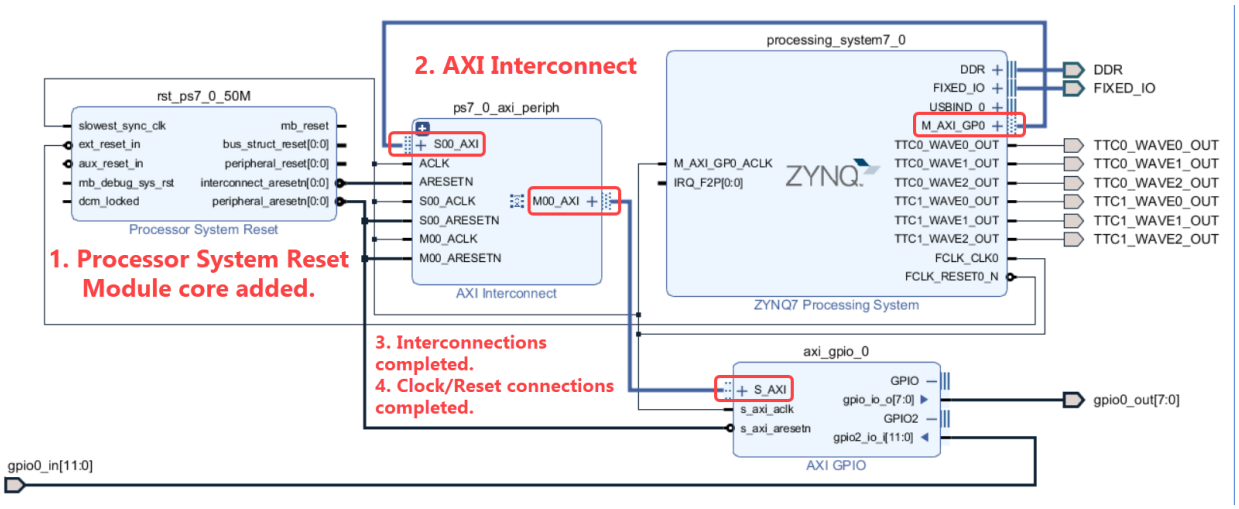


Figure 38. The updated block diagram is as shown.

2.2.5.4 Validate Design

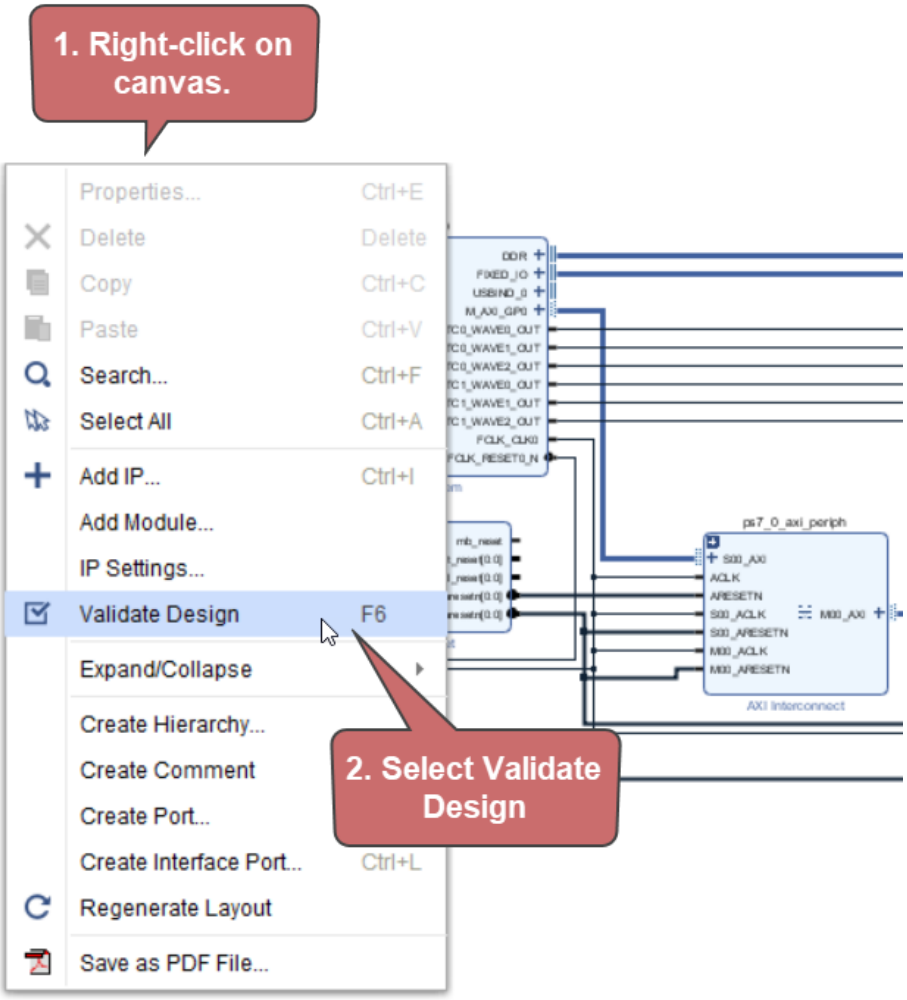


Figure 39. Validate the design.

Figure 40. Click OK to continue.

2.2.6 Add AXI Quad SPI IP

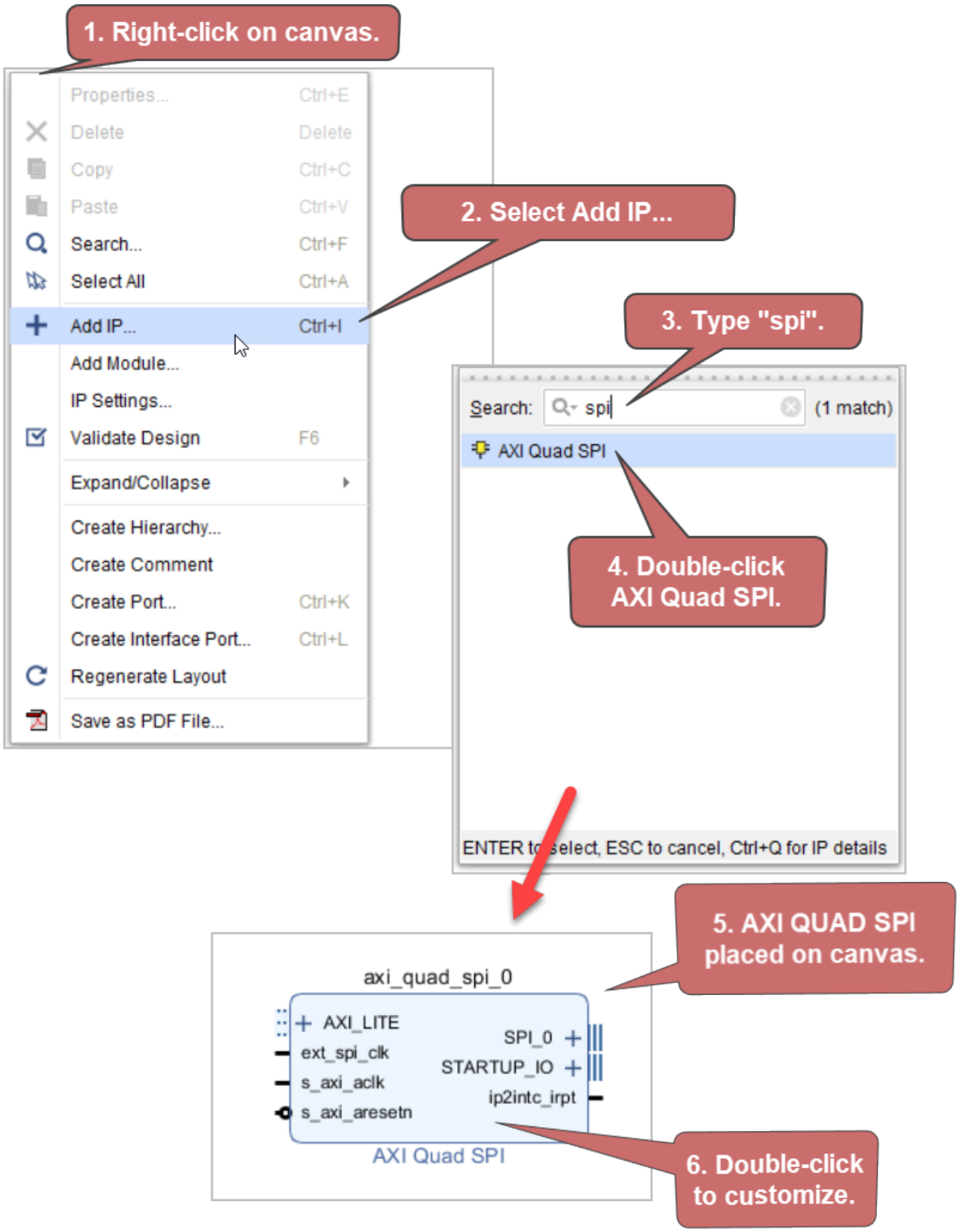


Figure 41. Add the AXI Quad SPI module.

2.2.6.1 Customise AXI Quad SPI IP

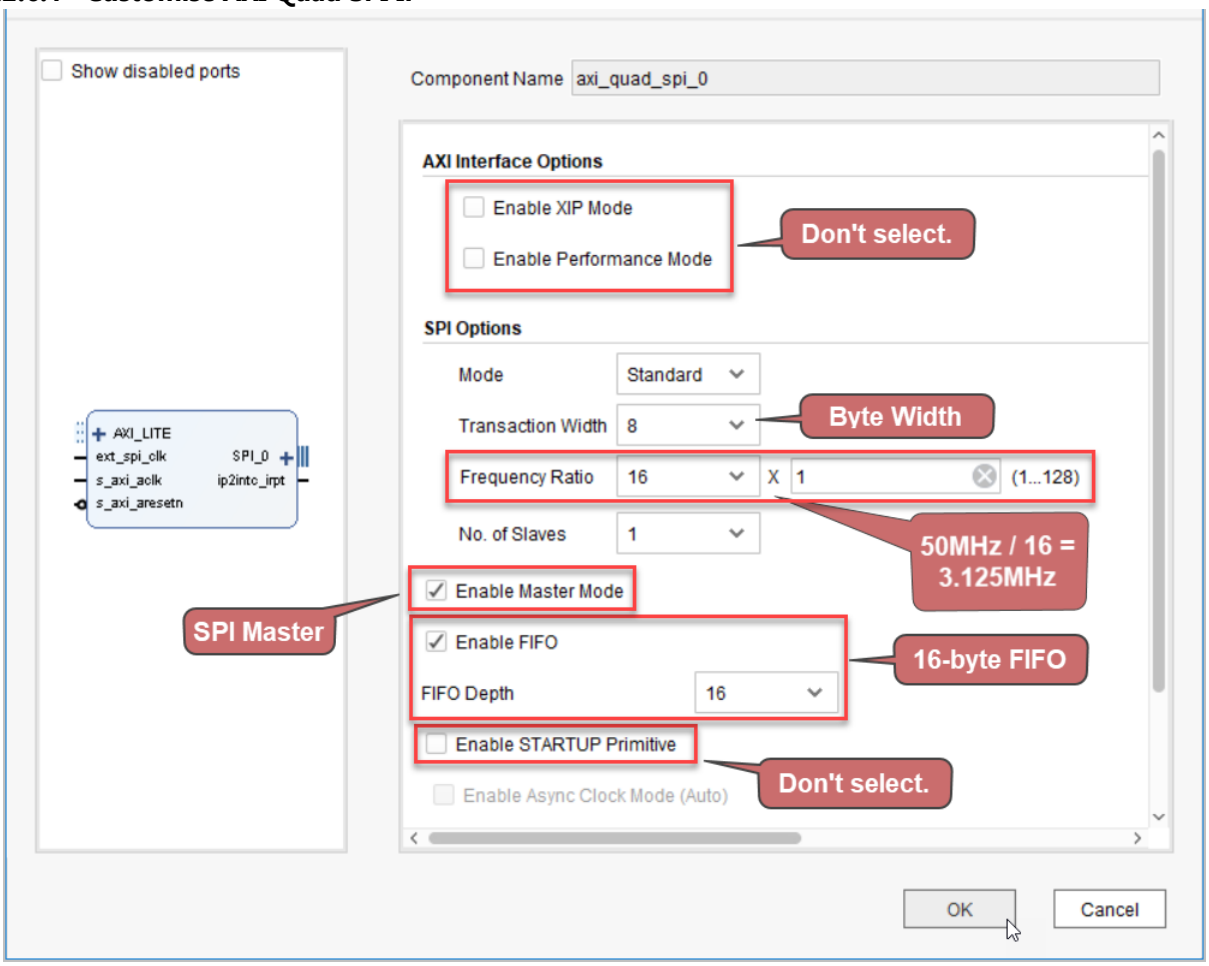


Figure 42. Customise the AXI Quad SPI as shown and click OK.

2.2.6.2 Add SPI Input/Output Pins

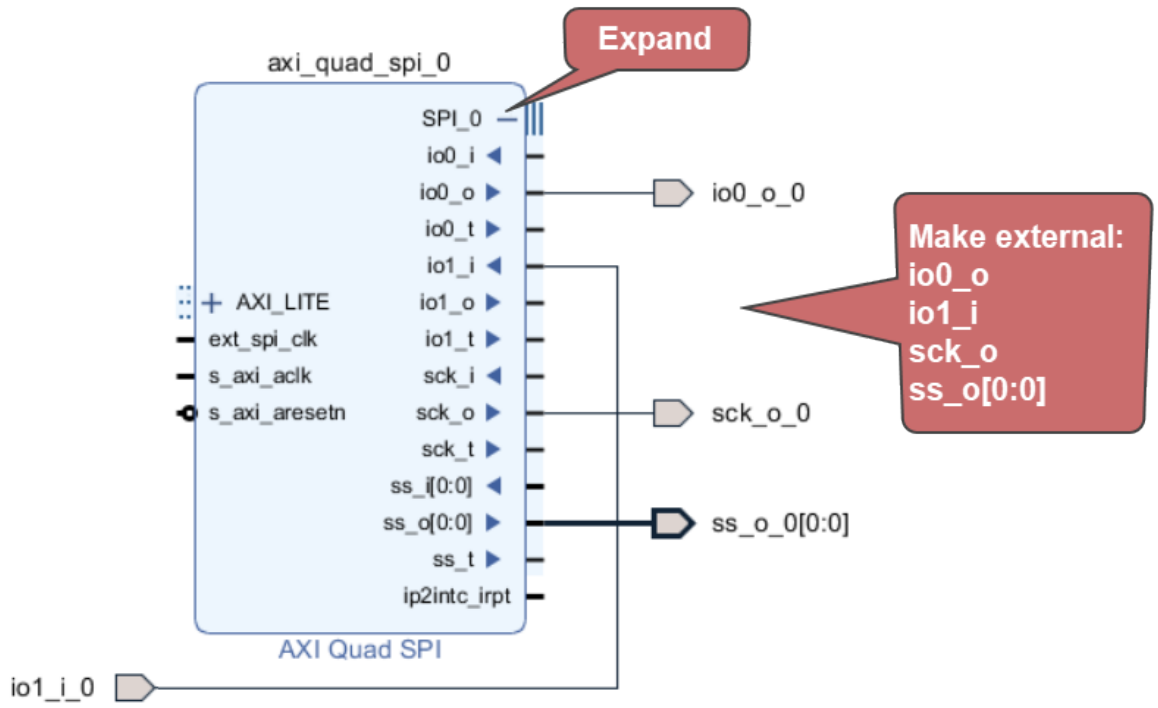


Figure 43. Add individual SPI pins

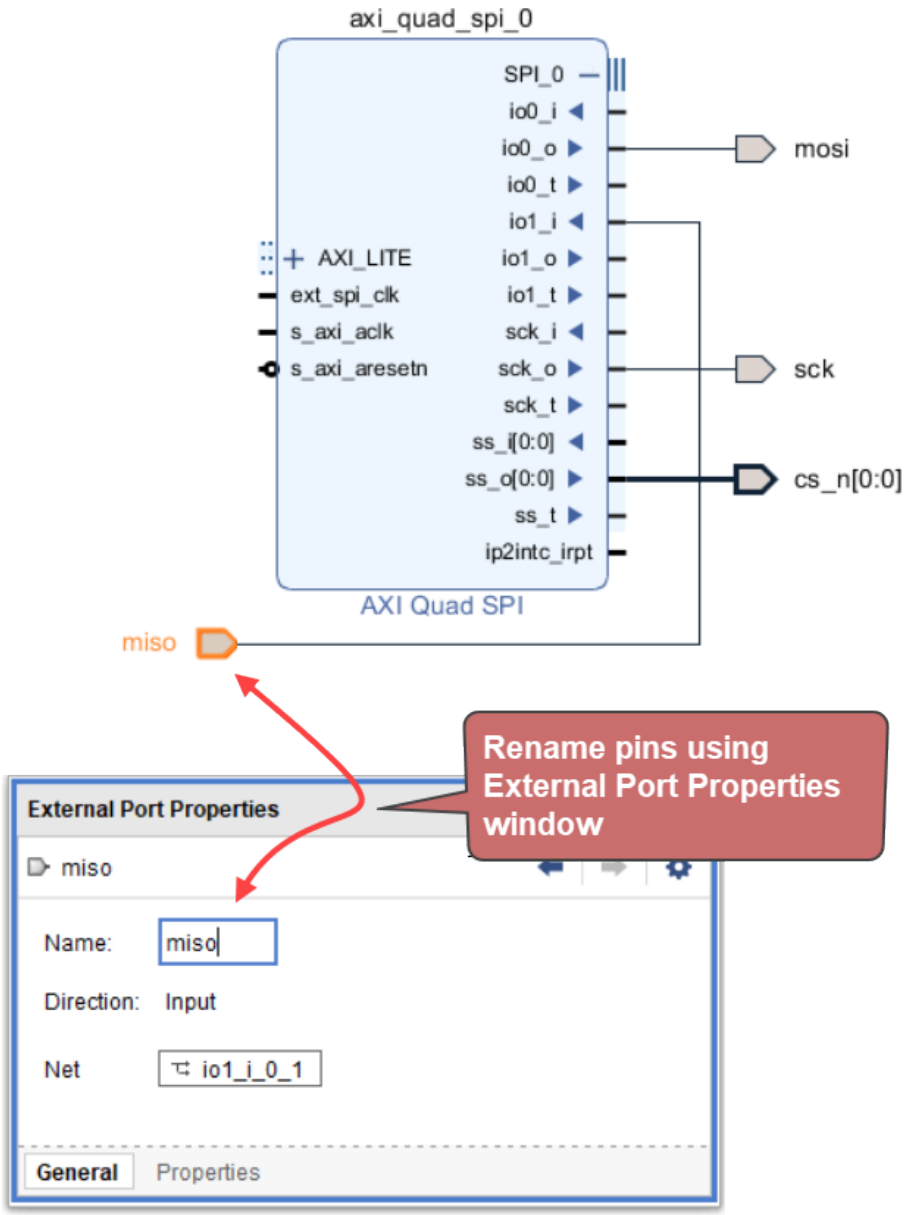


Figure 44. Rename the pins to match standard SPI names.

2.2.6.3 Run Connection Automation to Update Block Design

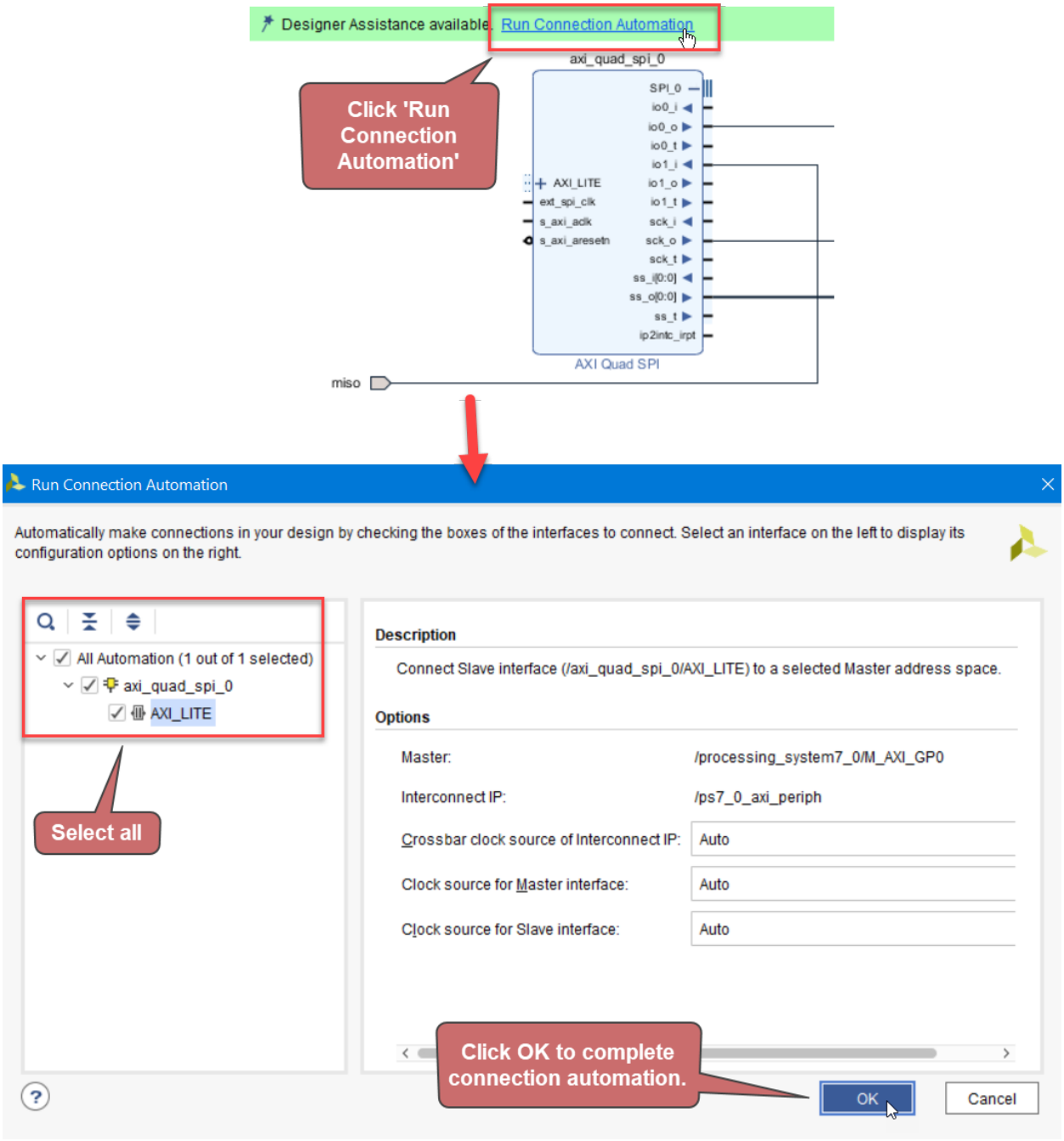


Figure 45. Run Connection Automation again.

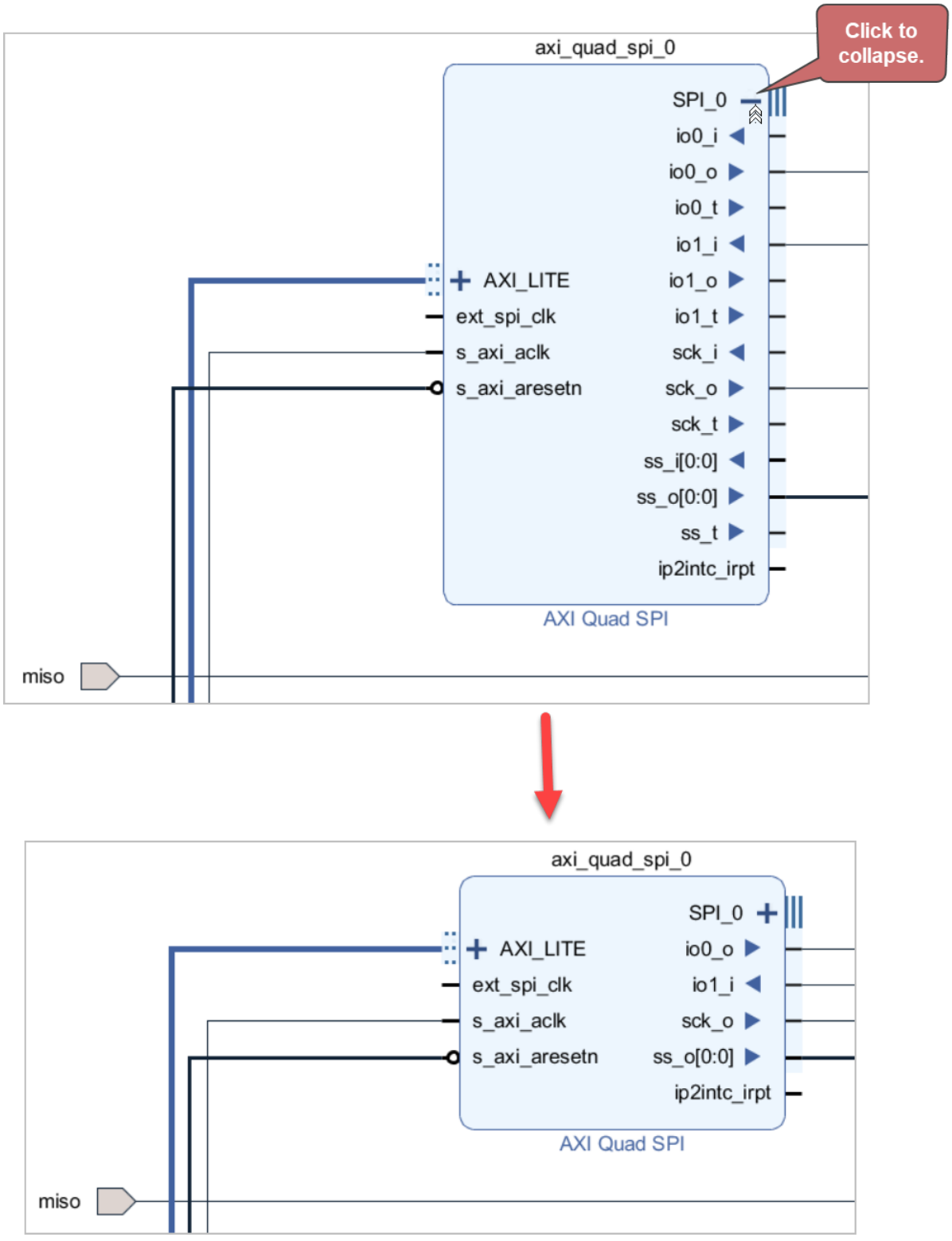


Figure 46. Tidy up block diagram by collapsing SPI_0 port.

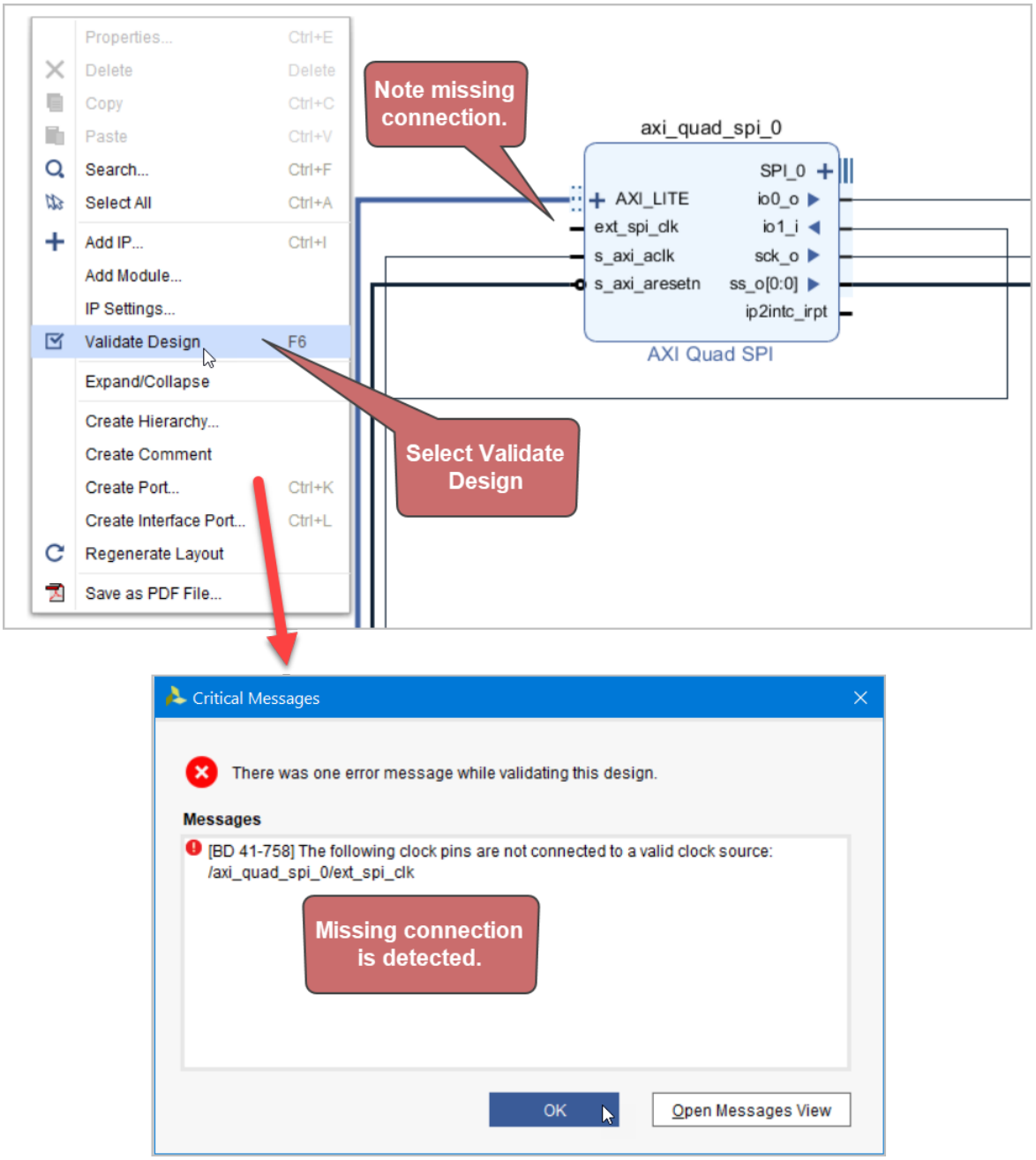


Figure 47. Validate the design; an error should be flagged.

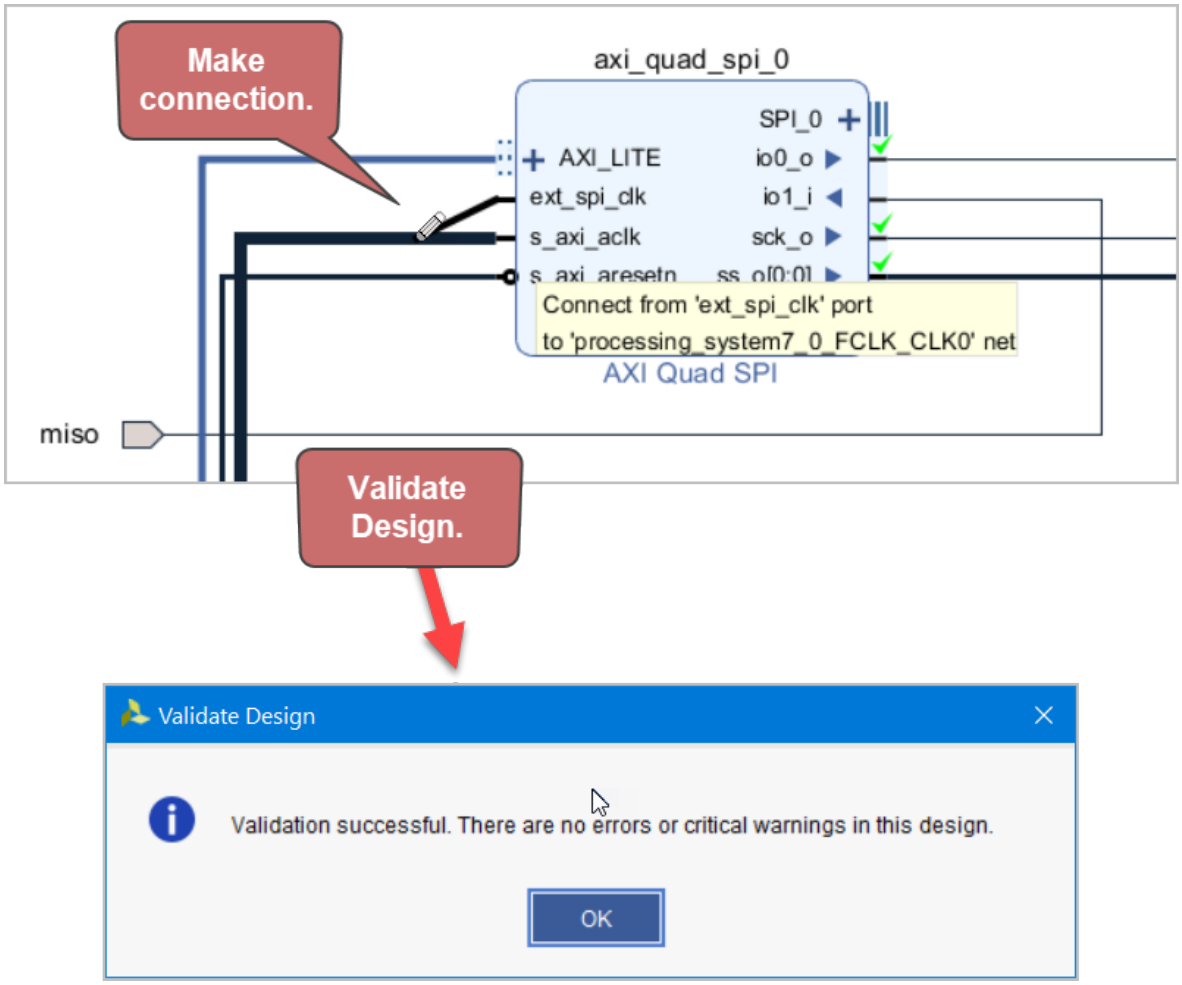


Figure 48. Fix the error by connecting ext_spi_clk to s_axi_aclk.

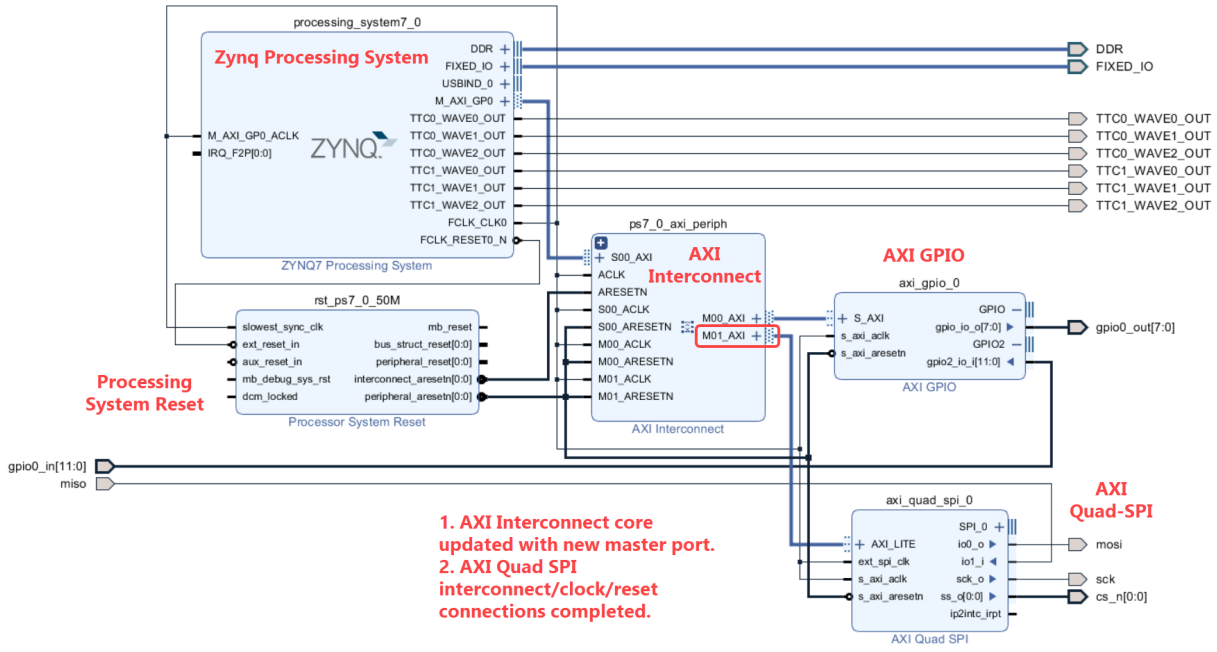


Figure 49. Block diagram at this stage of the design.

2.2.7 Add Concat IP for PmodACL Interrupts

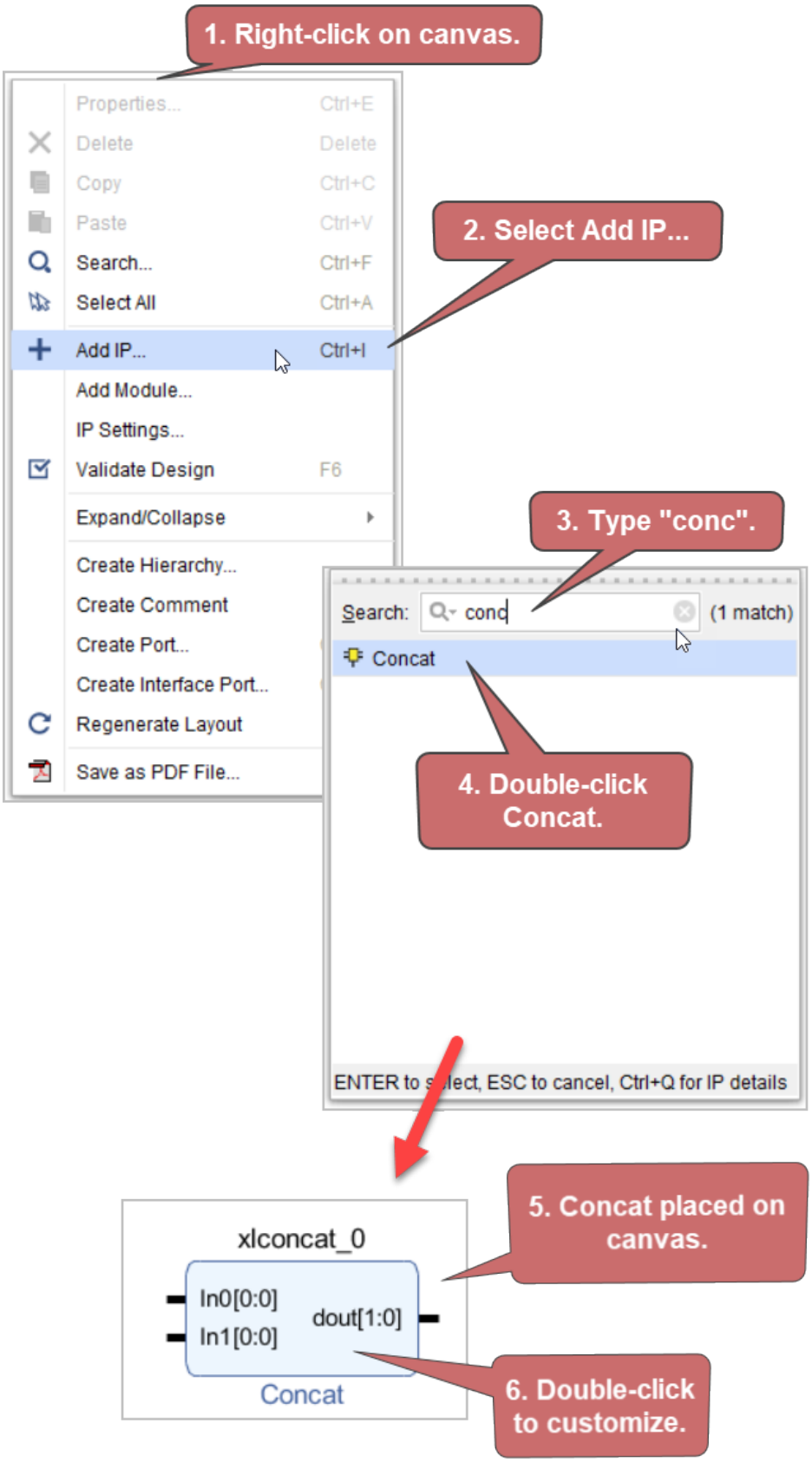


Figure 50. Add the Concat Utility IP for interrupt logic.

2.2.7.1 Customise Concat IP

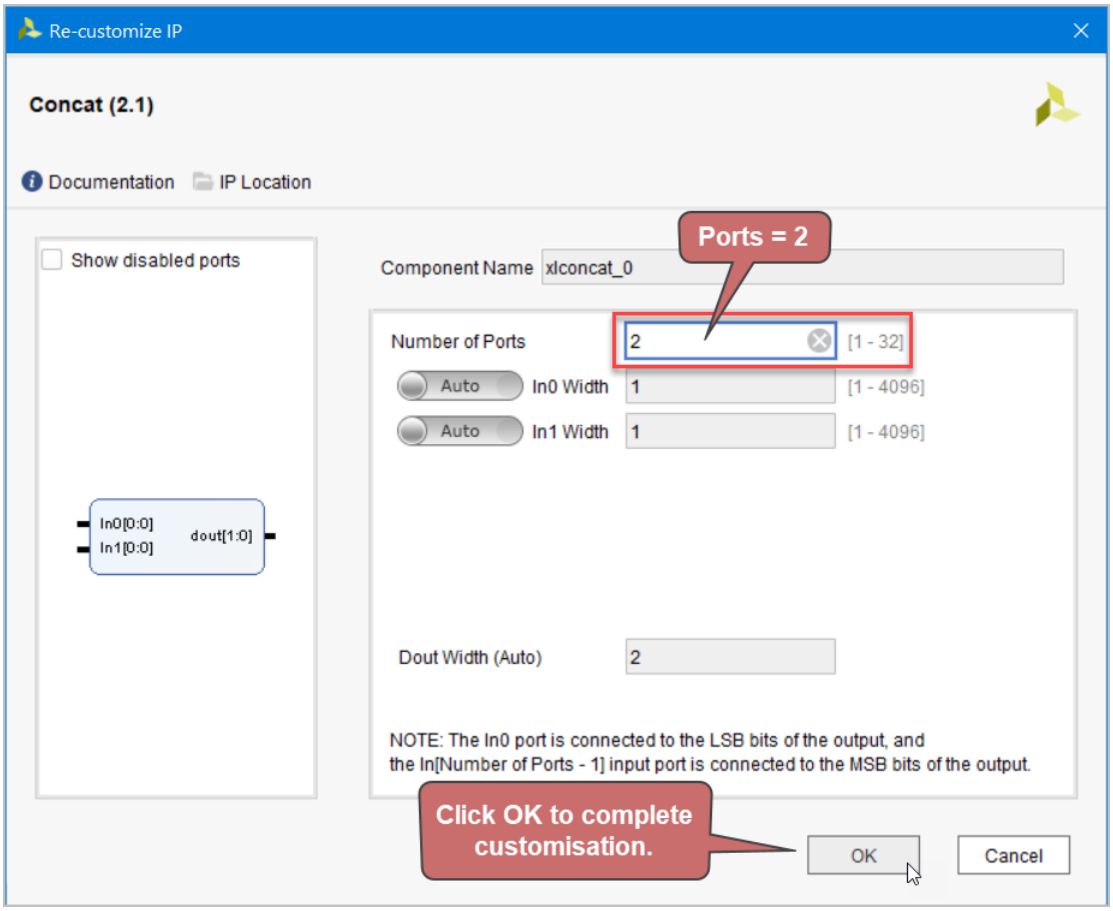


Figure 51. Ensure that two ports are selected and click OK

2.2.7.2 Add PmodACL Interrupt Pins

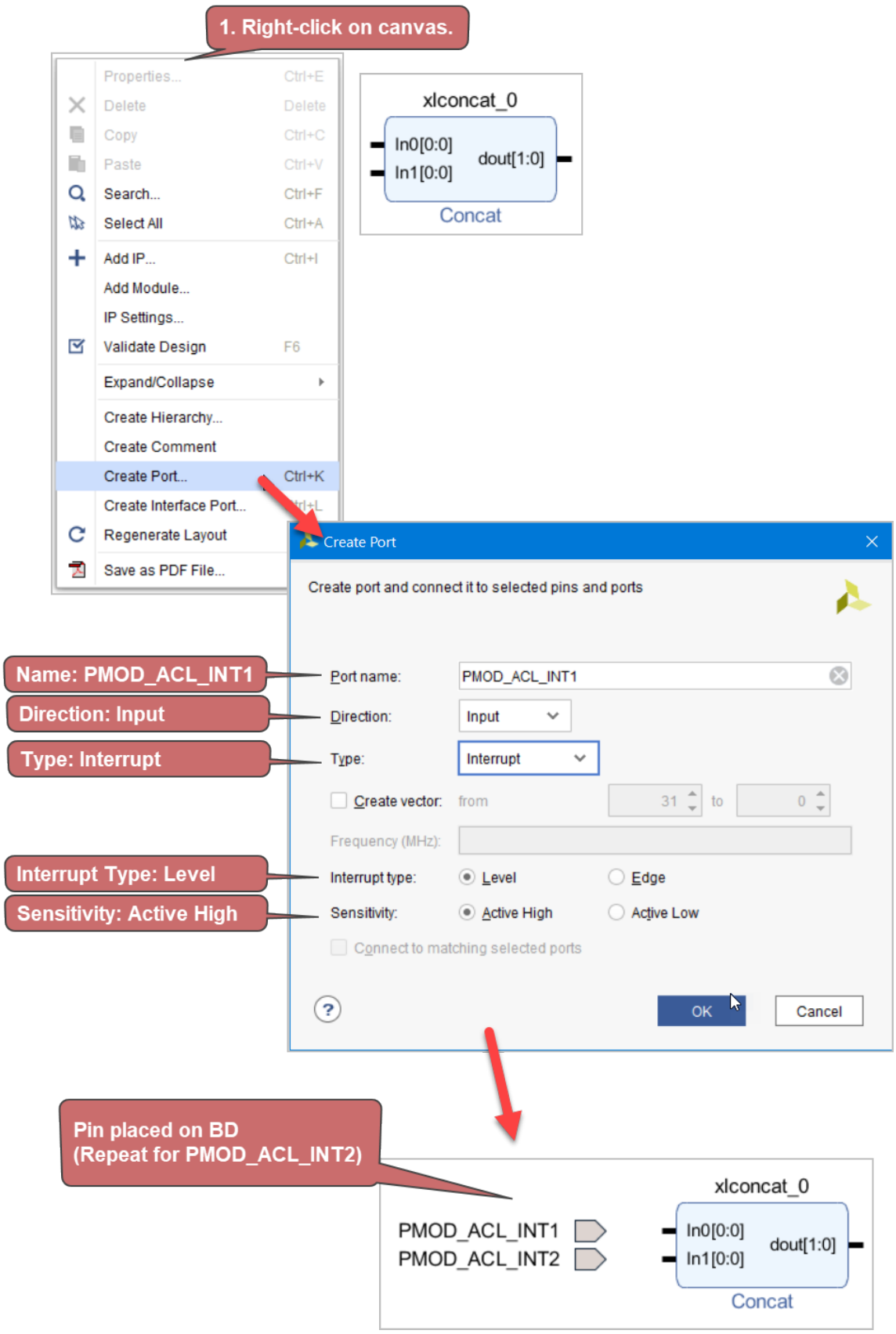


Figure 52. Add the Pmod ACL interrupt pins.

2.2.7.3 Make PmodACL Interrupt Pin Connections

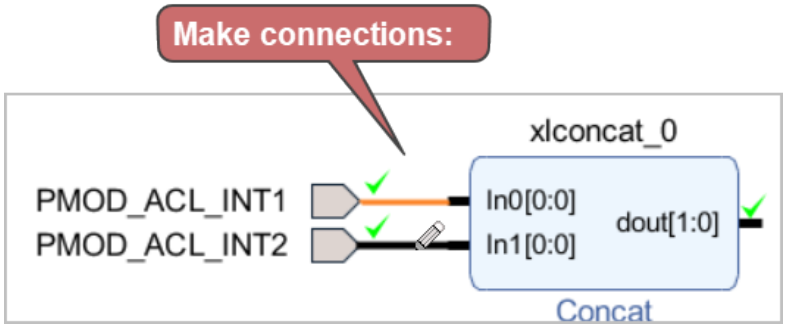


Figure 53. Make the interrupt pin connections.

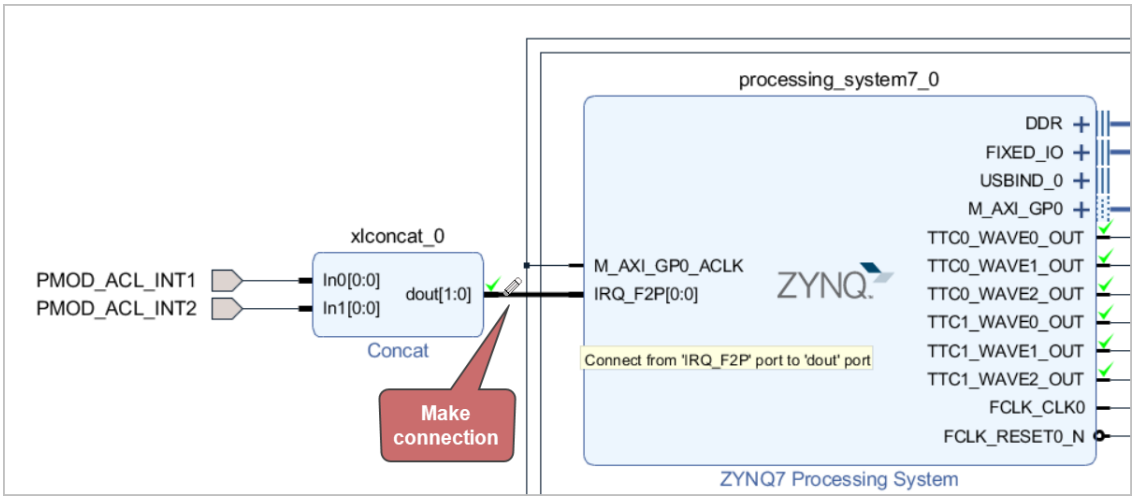


Figure 54. Connect the Concat IP output to IRQ_F2P port on the Zynq IP.

2.2.7.4 Validate Design

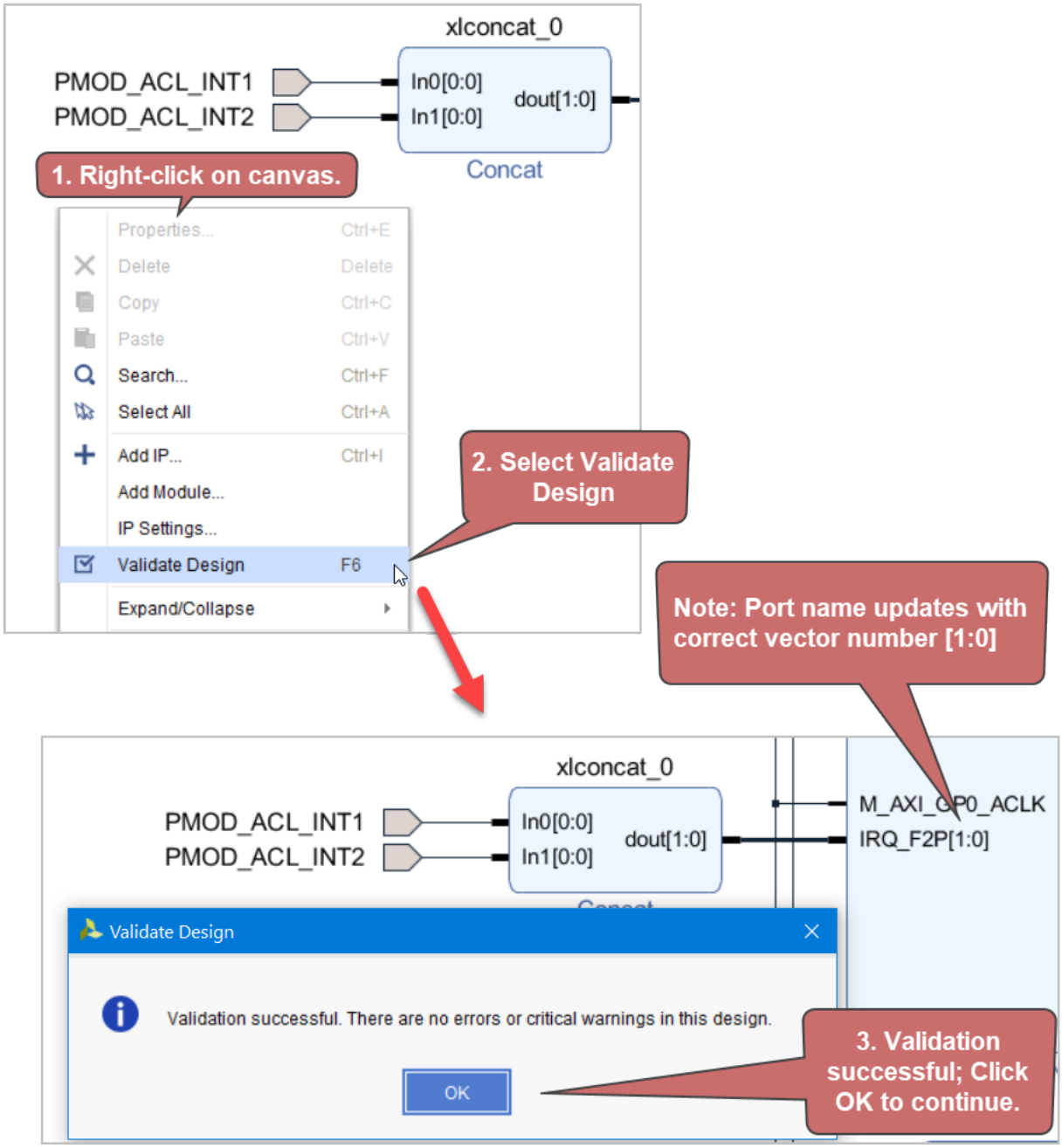


Figure 55. Validate the design for the final time.

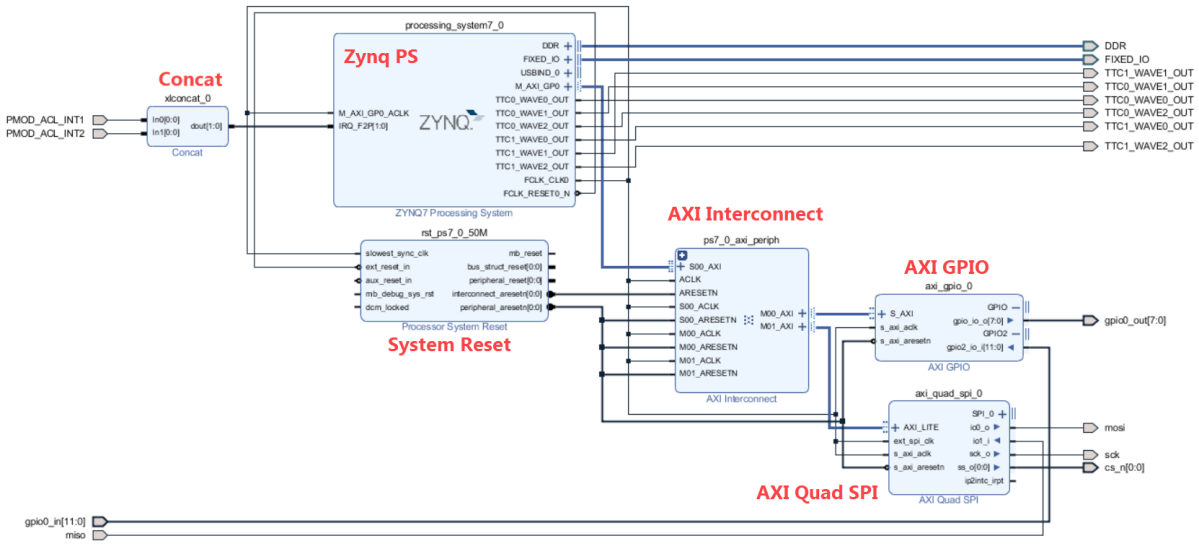


Figure 56. The final block diagram.

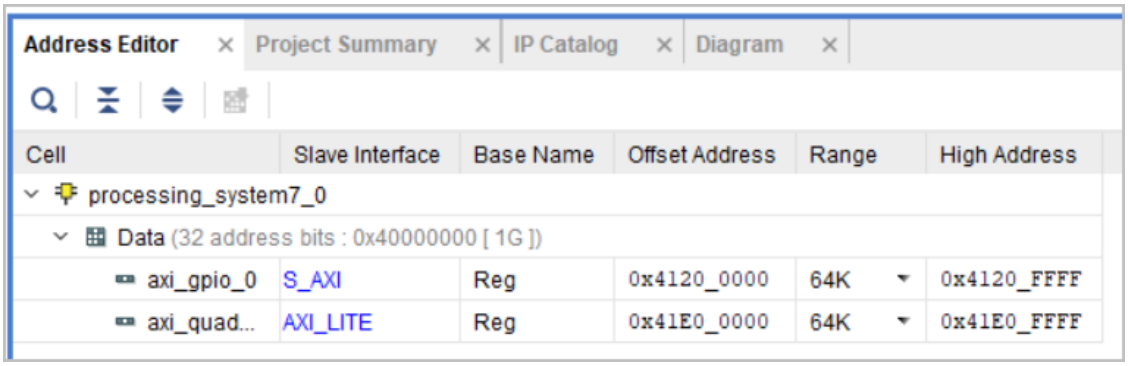


Figure 57. The address information for the AXI GPIO and AXI Quad IP can be viewed in the Address Editor tab.

2.3 Finalize the Design

2.3.1 Generate Output Products

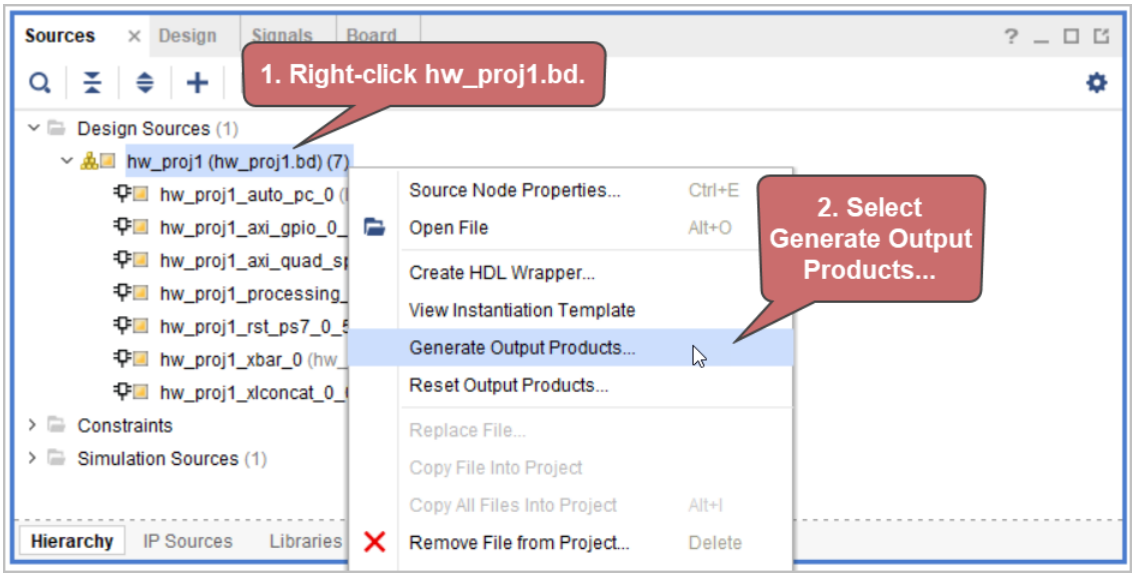


Figure 58. Select Generate Output Products...

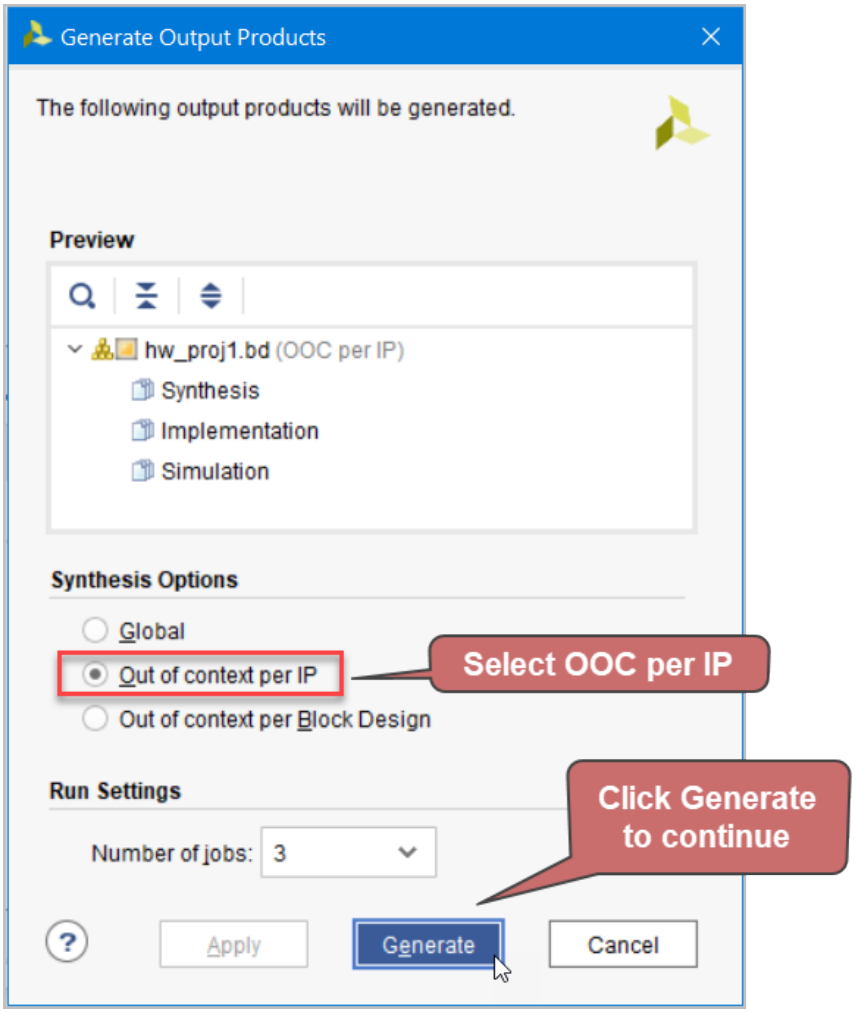


Figure 59. Ensure OOC per IP is selected, and click Generate.

Figure 60. OOC can be set to run in the background.

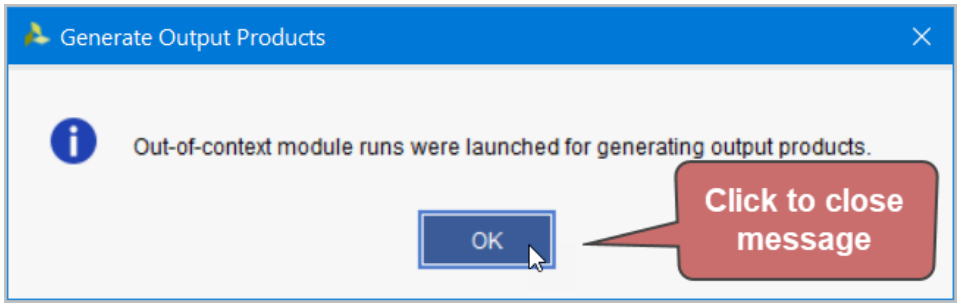


Figure 61. Click OK when the Generate Output Products dialog appears.

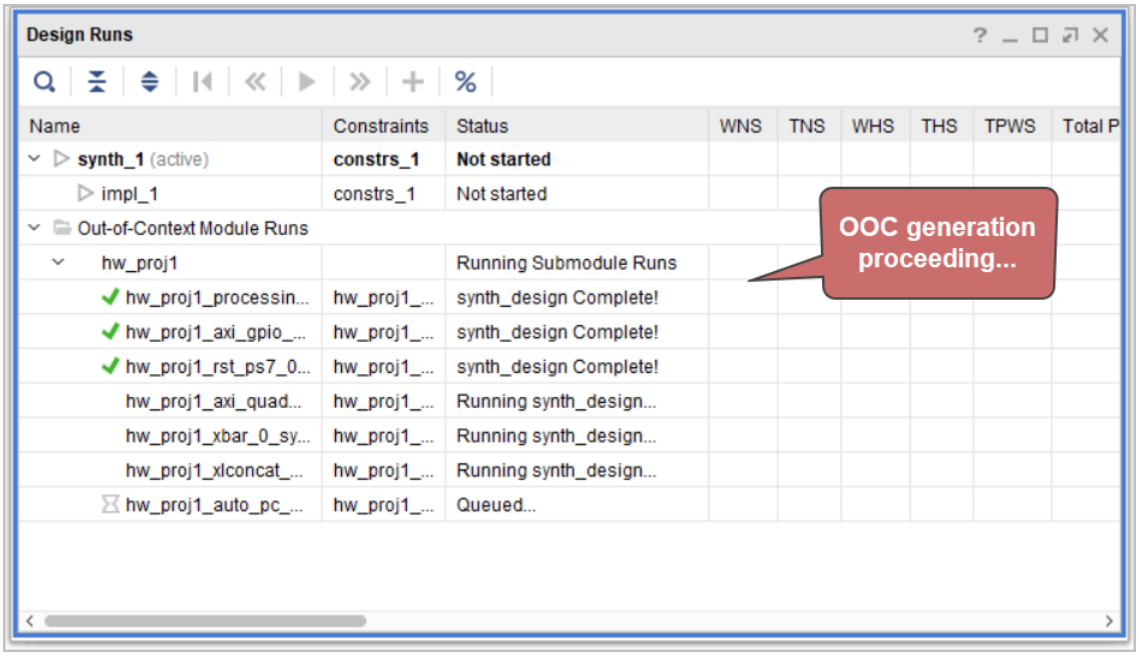


Figure 62. OOC generation runs...

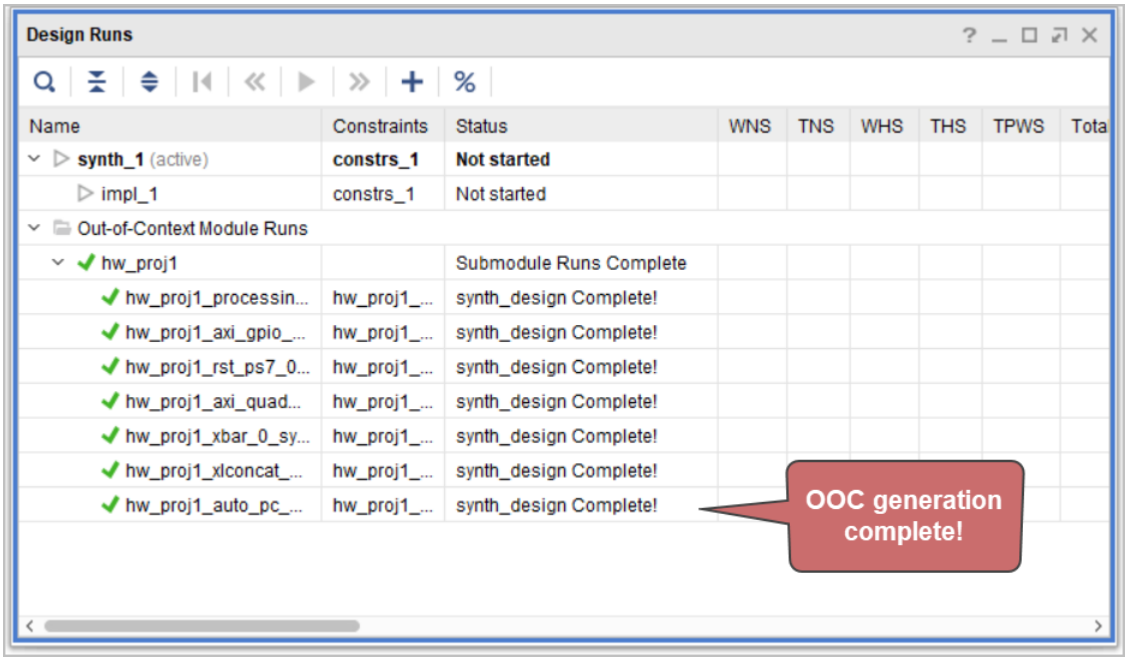


Figure 63. OOC Generation is complete.

2.3.2 Create HDL Wrapper

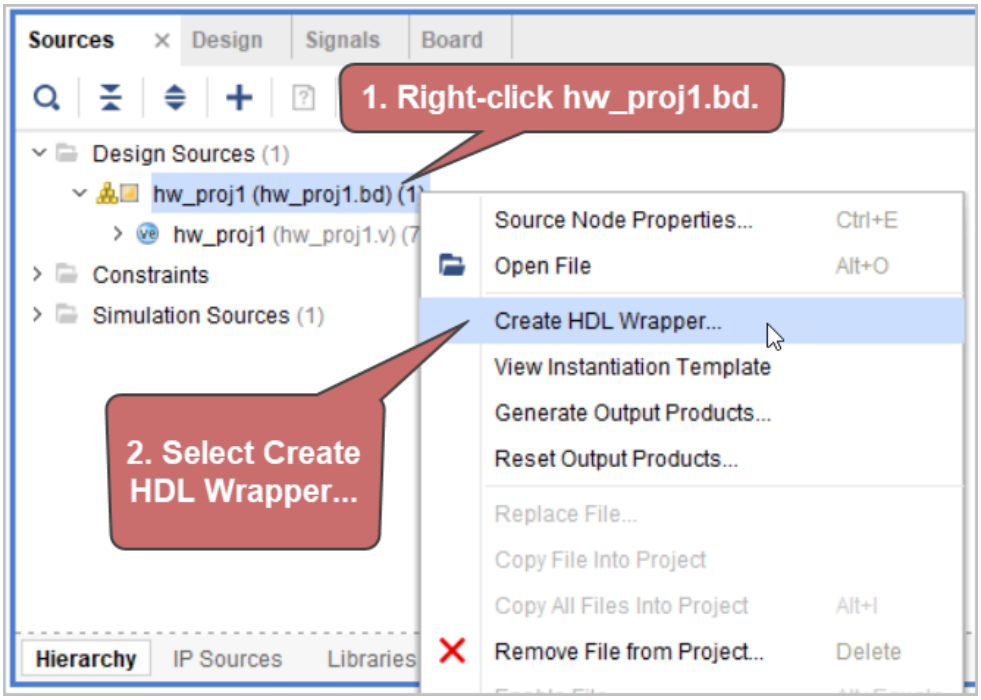


Figure 64. Create the HDL top-level wrapper.

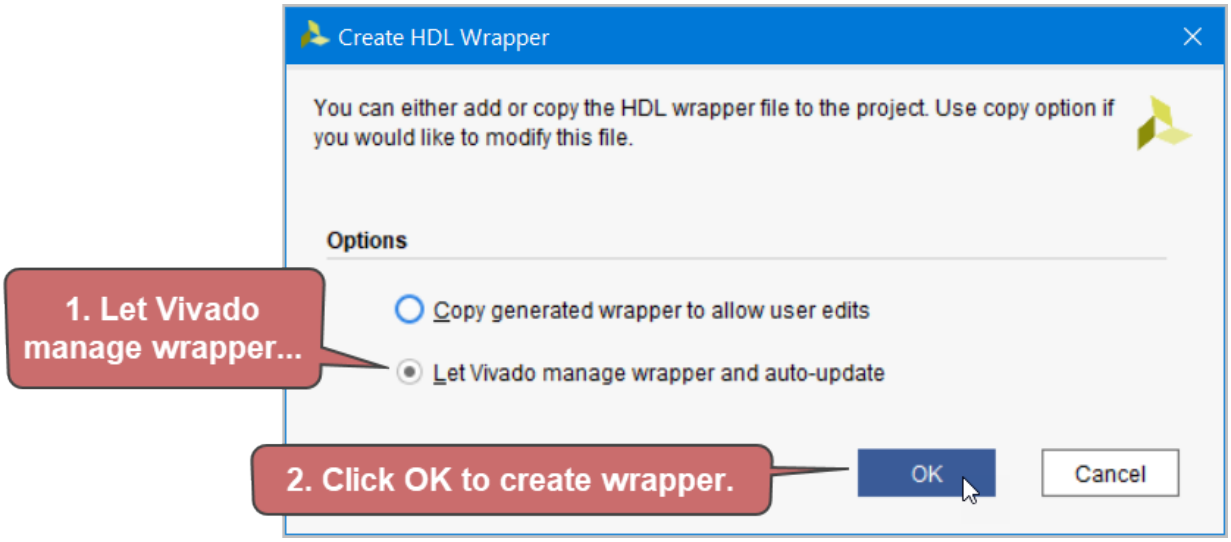


Figure 65. Let Vivado manage the wrapper, and click OK to continue.

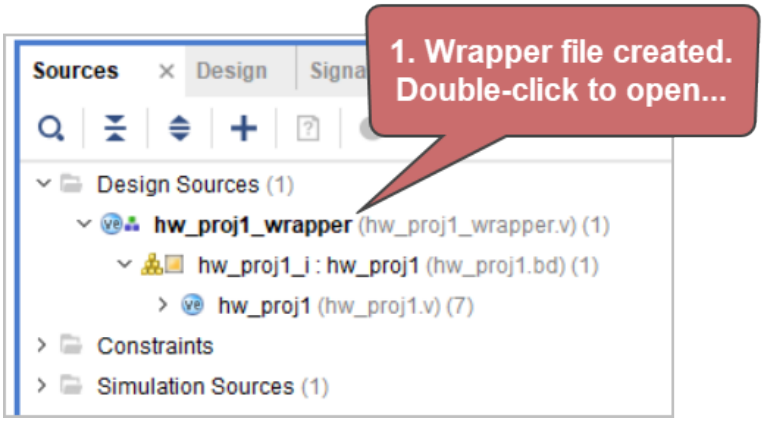


Figure 66. When created, double-click the wrapper to inspect it.

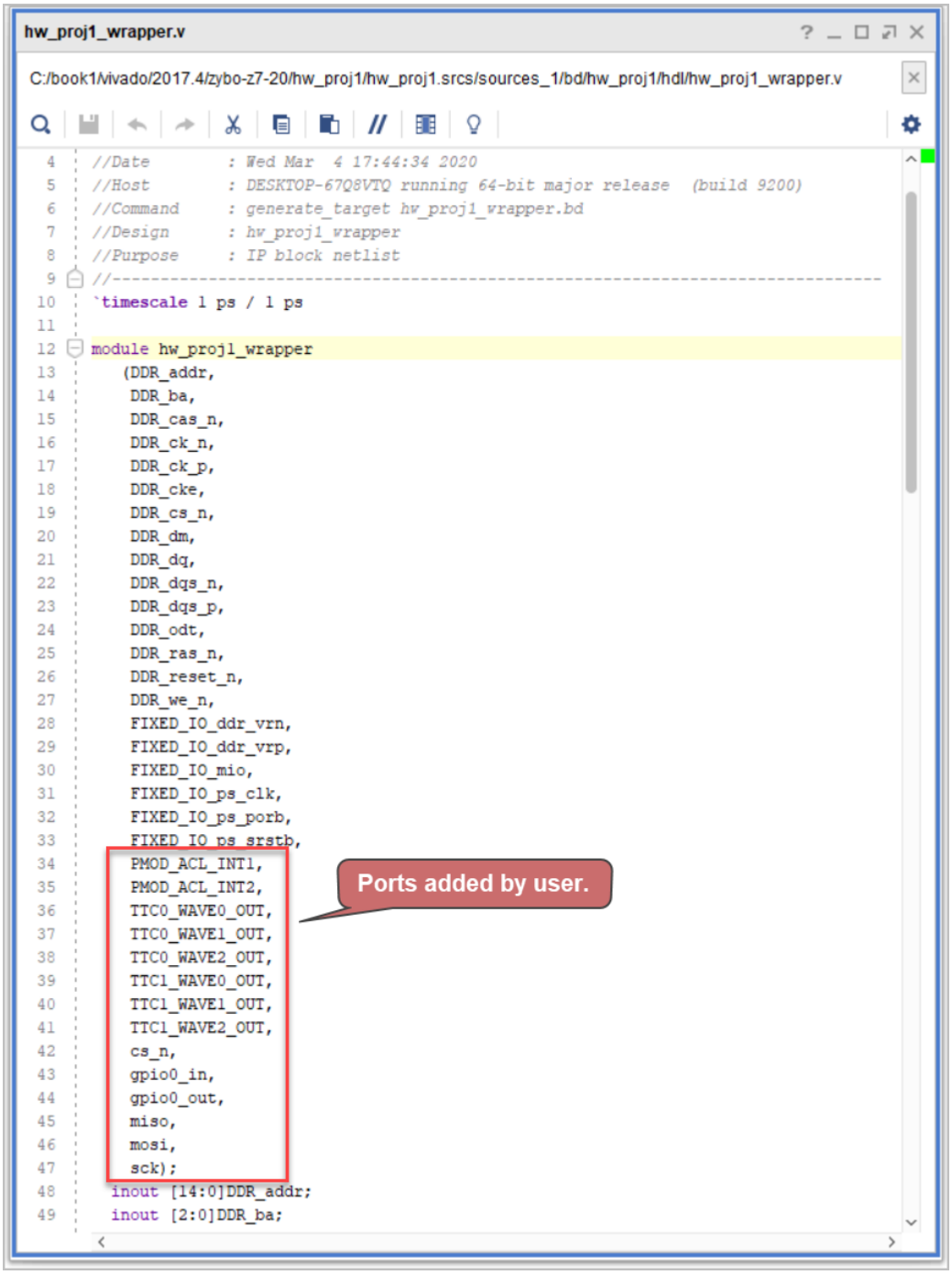


Figure 67. Check that the ports added by the user are present.

2.3.3 Add Constraints for the Project

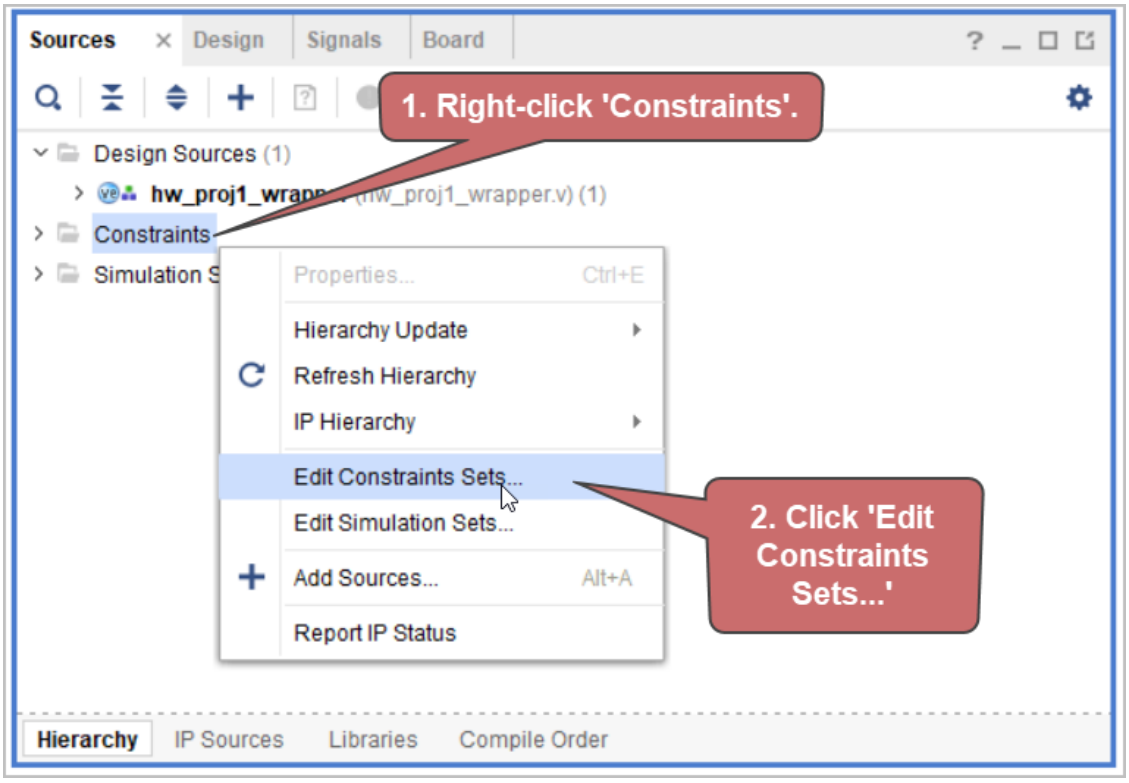


Figure 68. Abc

There are two options to create the constraints file. The first option is simply to copy the constraints file into the project (Section 2.3.3.1). The second is to create a new file and add the constraints manually (Section 2.3.3.2). Each is covered in the coming pages.

2.3.3.1 Option 1: Copy the constraints into the project

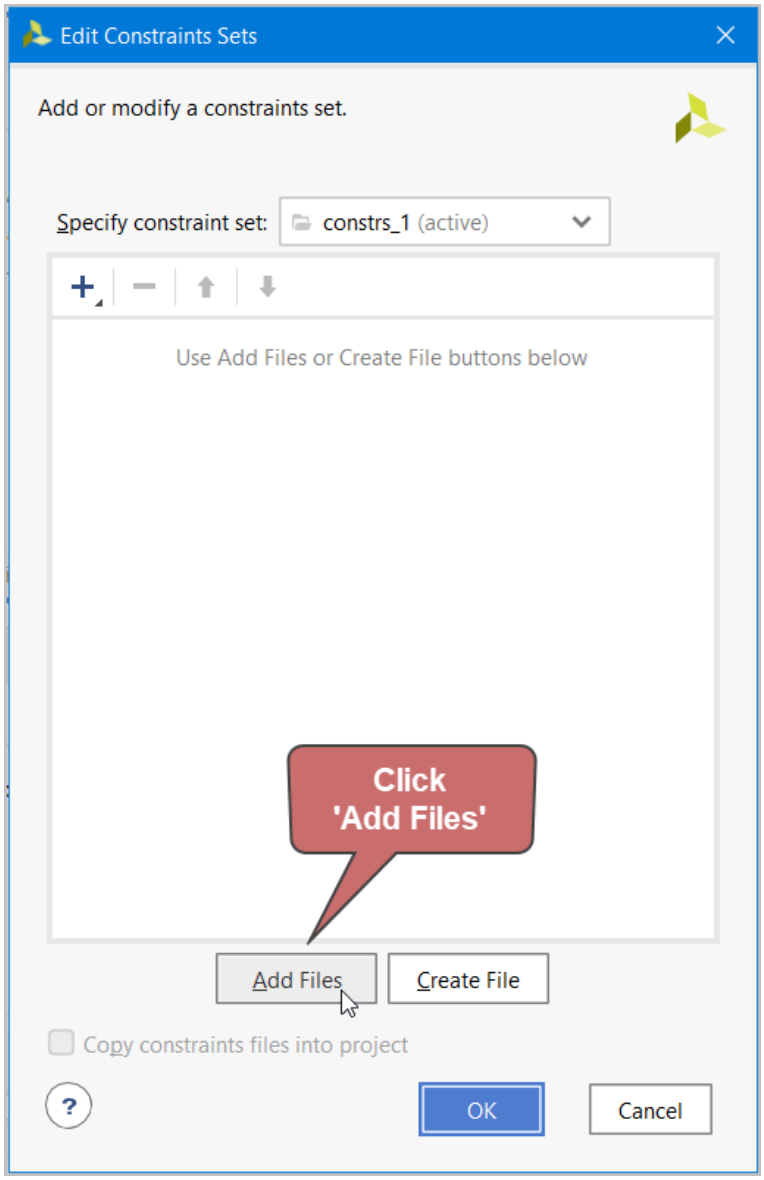


Figure 69. Click Add Files

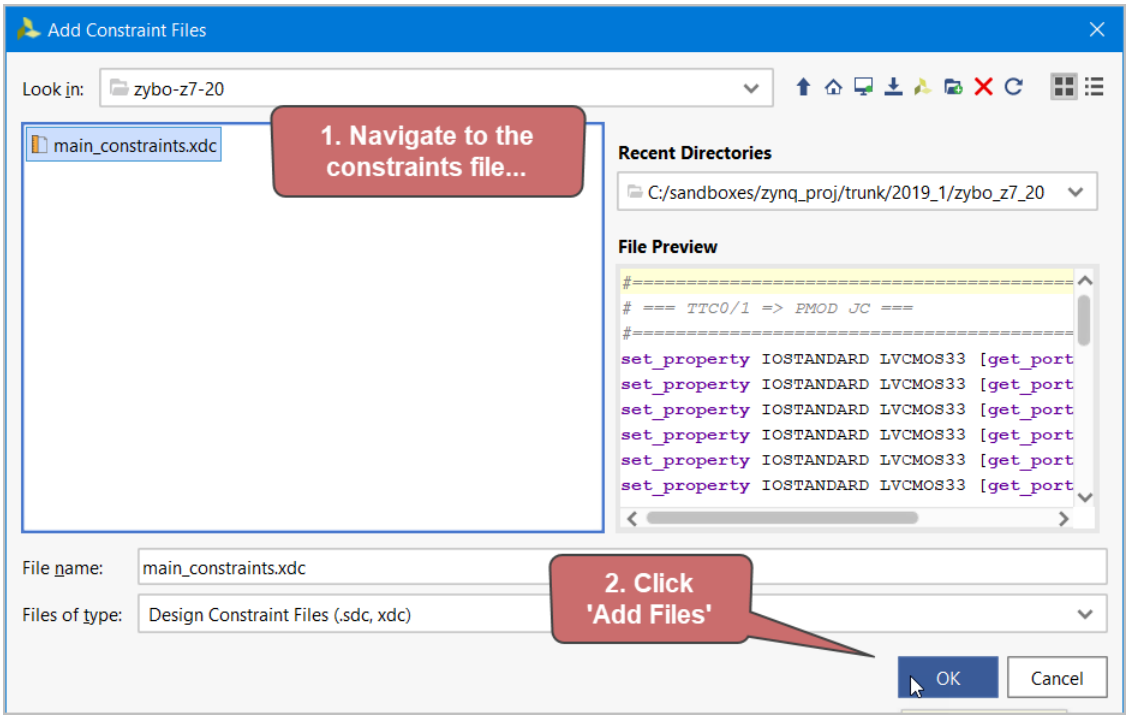


Figure 70. Browse to the supplied constraints file, and click OK to add it.

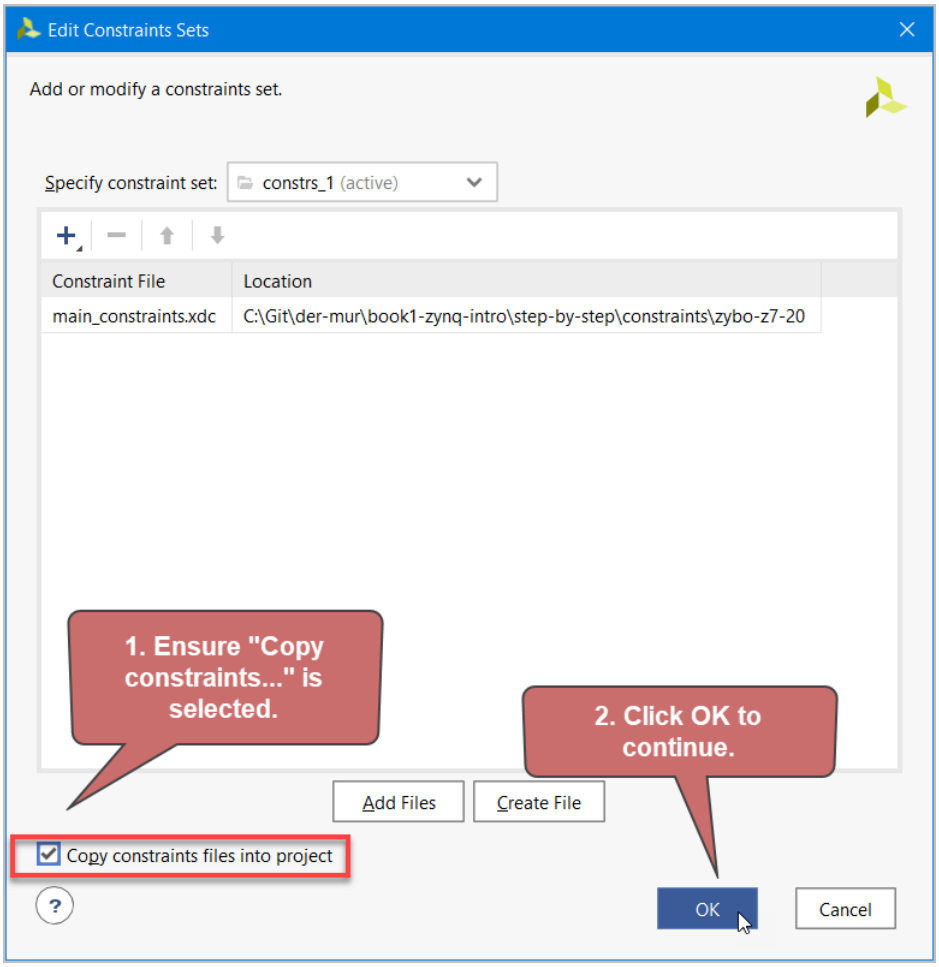


Figure 71. Click OK to continue

2.3.3.2 Option 2: Create the File and Add Constraints

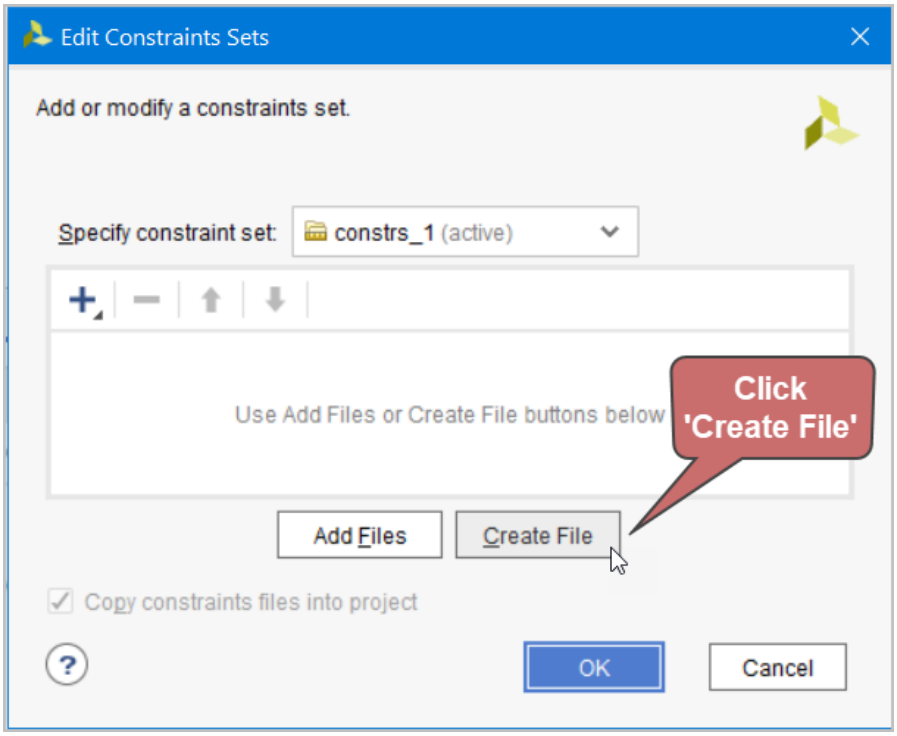


Figure 72. Create the file.

CHECK THIS SECTION WITH 2018.3, 2019.1

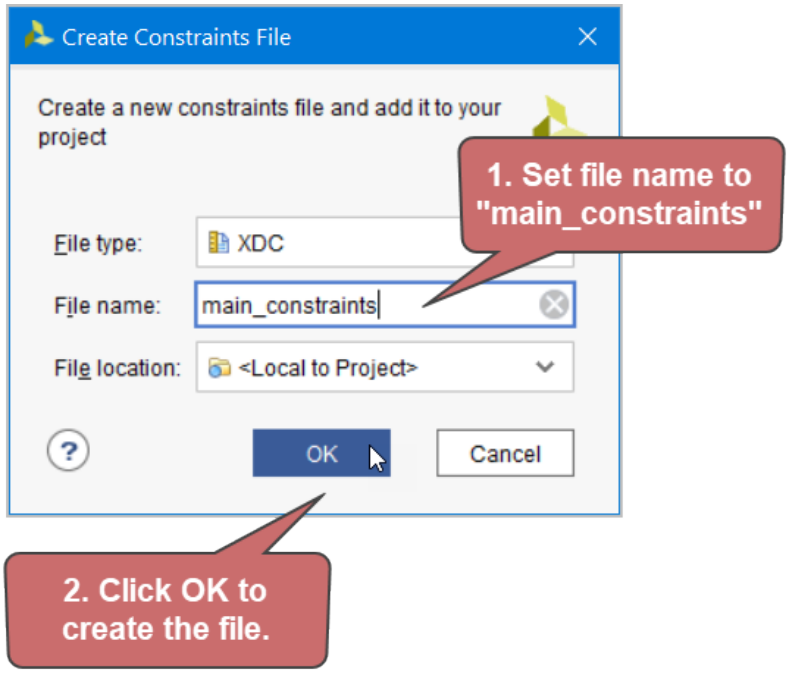


Figure 73. Name the file to "main_constraints" and click OK.

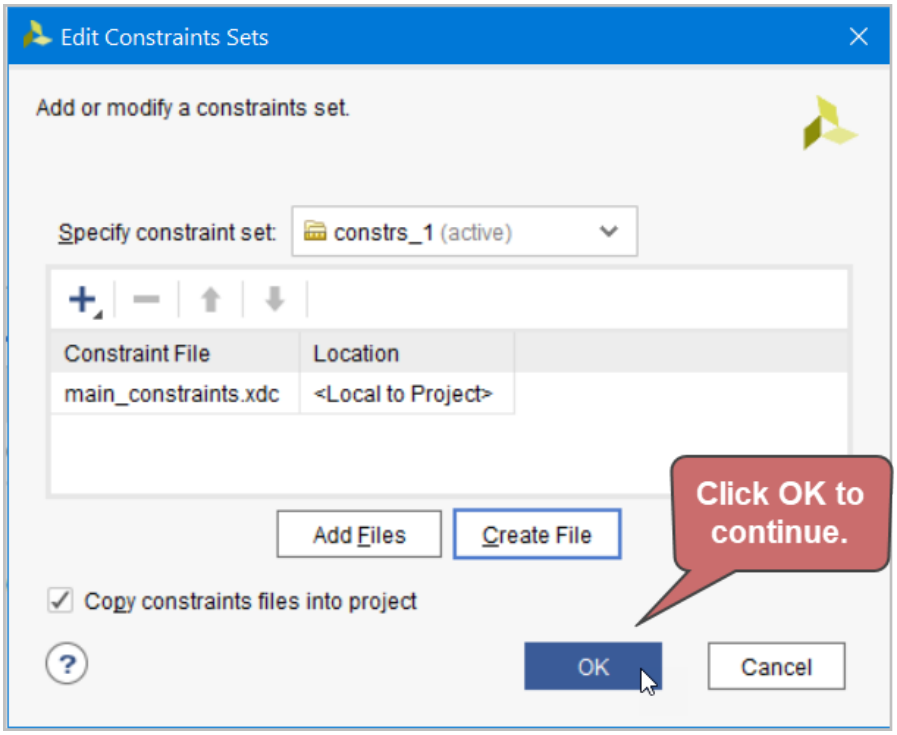


Figure 74. Click OK to add it to the project.

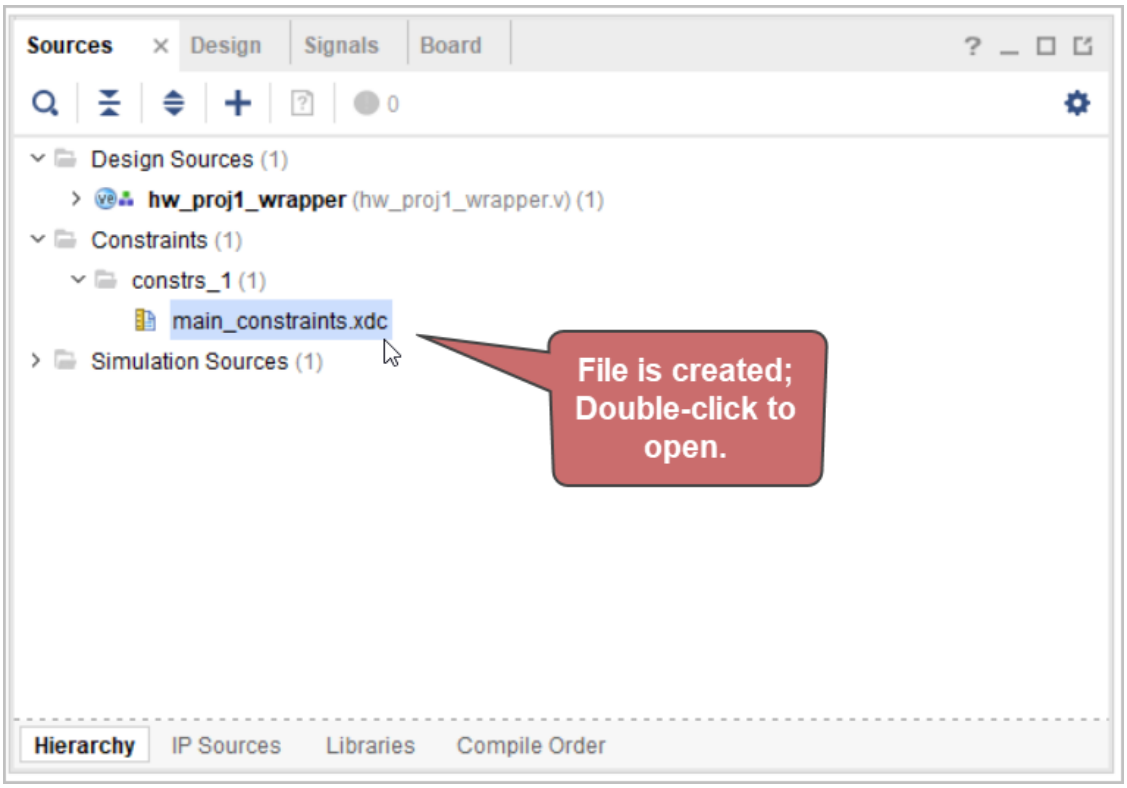


Figure 75. Open the constraints file.

Copy in the constraints as shown in the next two pages.

```
#============================================================#
# === AXI GPIO0 ===
#============================================================#
set_property IOSTANDARD LVCMOS33 [get_ports gpio0_out*]
set_property IOSTANDARD LVCMOS33 [get_ports gpio0_in*]

#===============
# gpio0_out[7:0]
#===============
# Board LEDs: LD0 - LD3
set_property PACKAGE_PIN M14 [get_ports {gpio0_out[0]}]
set_property PACKAGE_PIN M15 [get_ports {gpio0_out[1]}]
set_property PACKAGE_PIN G14 [get_ports {gpio0_out[2]}]
set_property PACKAGE_PIN D18 [get_ports {gpio0_out[3]}]

# PMOD JE Pins 1-4
set_property PACKAGE_PIN V12 [get_ports {gpio0_out[4]}]
set_property PACKAGE_PIN W16 [get_ports {gpio0_out[5]}]
set_property PACKAGE_PIN J15 [get_ports {gpio0_out[6]}]
set_property PACKAGE_PIN H15 [get_ports {gpio0_out[7]}]

#===============
# gpio0_in[11:0]
#===============
# Board Push-buttons: BTN0 - BTN3
set_property PACKAGE_PIN K18 [get_ports {gpio0_in[0]}]
set_property PACKAGE_PIN P16 [get_ports {gpio0_in[1]}]
set_property PACKAGE_PIN K19 [get_ports {gpio0_in[2]}]
set_property PACKAGE_PIN Y16 [get_ports {gpio0_in[3]}]

# Board Switches: SW0 - SW3
set_property PACKAGE_PIN G15 [get_ports {gpio0_in[4]}]
set_property PACKAGE_PIN P15 [get_ports {gpio0_in[5]}]
set_property PACKAGE_PIN W13 [get_ports {gpio0_in[6]}]
set_property PACKAGE_PIN T16 [get_ports {gpio0_in[7]}]

# PMOD JE Pins 5-8
set_property PACKAGE_PIN V13 [get_ports {gpio0_in[8]}]
set_property PACKAGE_PIN U17 [get_ports {gpio0_in[9]}]
set_property PACKAGE_PIN T17 [get_ports {gpio0_in[10]}]
set_property PACKAGE_PIN Y17 [get_ports {gpio0_in[11]}]
```

```
#=========================================================#
# === TTC0/1 => PMOD JC ===
#=========================================================#
set_property IOSTANDARD LVCMOS33 [get_ports TTC0_WAVE0_OUT]
set_property IOSTANDARD LVCMOS33 [get_ports TTC0_WAVE1_OUT]
set_property IOSTANDARD LVCMOS33 [get_ports TTC0_WAVE2_OUT]
set_property IOSTANDARD LVCMOS33 [get_ports TTC1_WAVE0_OUT]
set_property IOSTANDARD LVCMOS33 [get_ports TTC1_WAVE1_OUT]
set_property IOSTANDARD LVCMOS33 [get_ports TTC1_WAVE2_OUT]

set_property PACKAGE_PIN V15 [get_ports TTC0_WAVE0_OUT]
set_property PACKAGE_PIN W15 [get_ports TTC0_WAVE1_OUT]
set_property PACKAGE_PIN T11 [get_ports TTC0_WAVE2_OUT]
# PIN 4 (T10) NOT USED
set_property PACKAGE_PIN W14 [get_ports TTC1_WAVE0_OUT]
set_property PACKAGE_PIN Y14 [get_ports TTC1_WAVE1_OUT]
set_property PACKAGE_PIN T12 [get_ports TTC1_WAVE2_OUT]
# PIN 10 (U12) NOT USED

#=========================================================#
# === PmodACL => PMOD JD ===
#=========================================================#
set_property IOSTANDARD LVCMOS33 [get_ports cs_n]
set_property IOSTANDARD LVCMOS33 [get_ports mosi]
set_property IOSTANDARD LVCMOS33 [get_ports miso]
set_property IOSTANDARD LVCMOS33 [get_ports sclk]
set_property IOSTANDARD LVCMOS33 [get_ports PMOD_ACL_INT2]
set_property IOSTANDARD LVCMOS33 [get_ports PMOD_ACL_INT1]

# cs_n => PMOD JD PIN 1 = T14
# mosi => PMOD JD PIN 2 = T15
# miso => PMOD JD PIN 3 = P14
# sclk => PMOD JD PIN 4 = R14
# PMOD_ACL_INT2 => PMOD JD PIN 7 = U14
# PMOD_ACL_INT1 => PMOD JD PIN 8 = U15
set_property PACKAGE_PIN T14 [get_ports cs_n]
set_property PACKAGE_PIN T15 [get_ports mosi]
set_property PACKAGE_PIN P14 [get_ports miso]
set_property PACKAGE_PIN R14 [get_ports sclk]
set_property PACKAGE_PIN U14 [get_ports PMOD_ACL_INT2]
set_property PACKAGE_PIN U15 [get_ports PMOD_ACL_INT1]
```

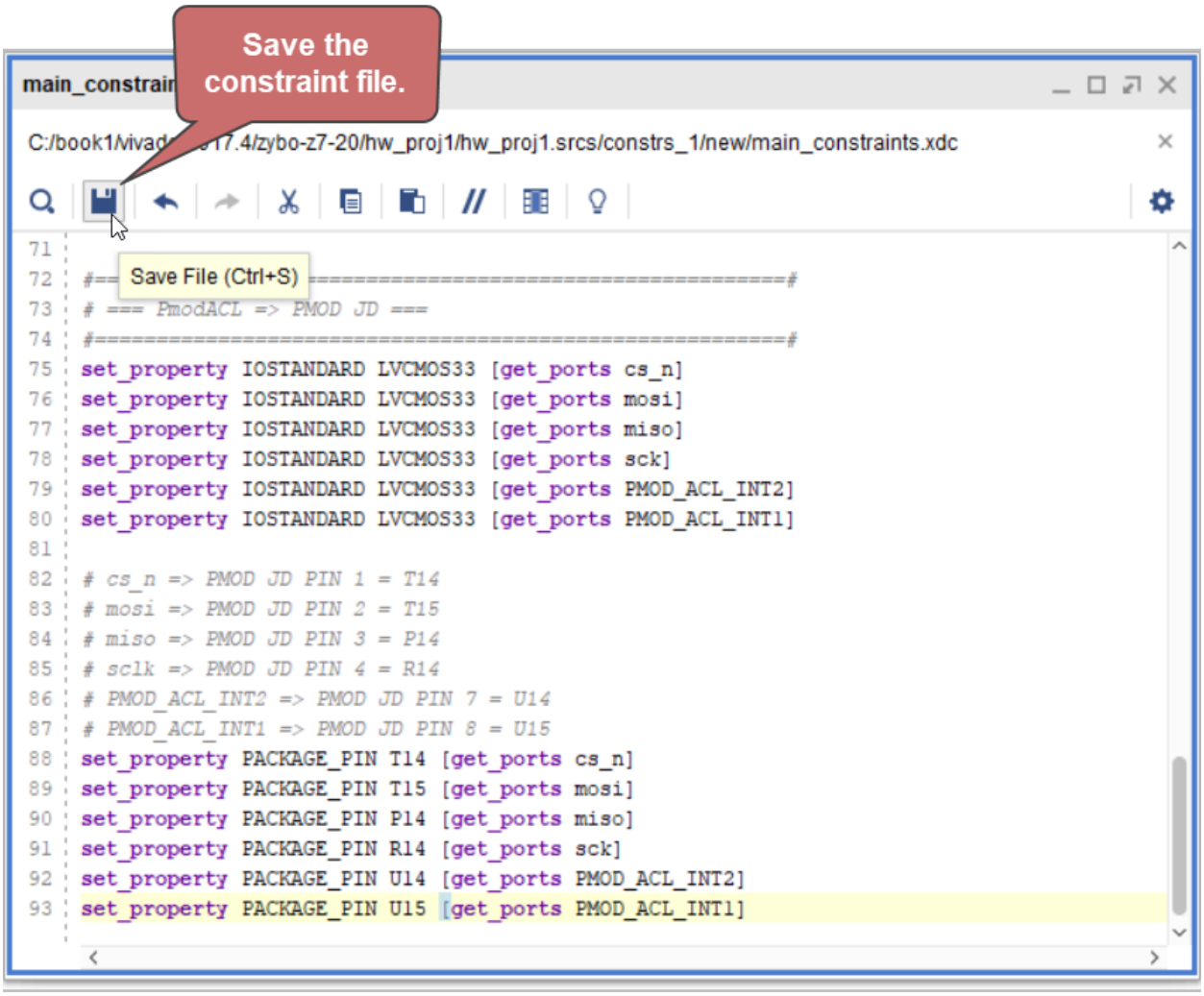


Figure 76. Save the constraints file.

2.4 Design Generation

2.4.1 Synthesize the Design

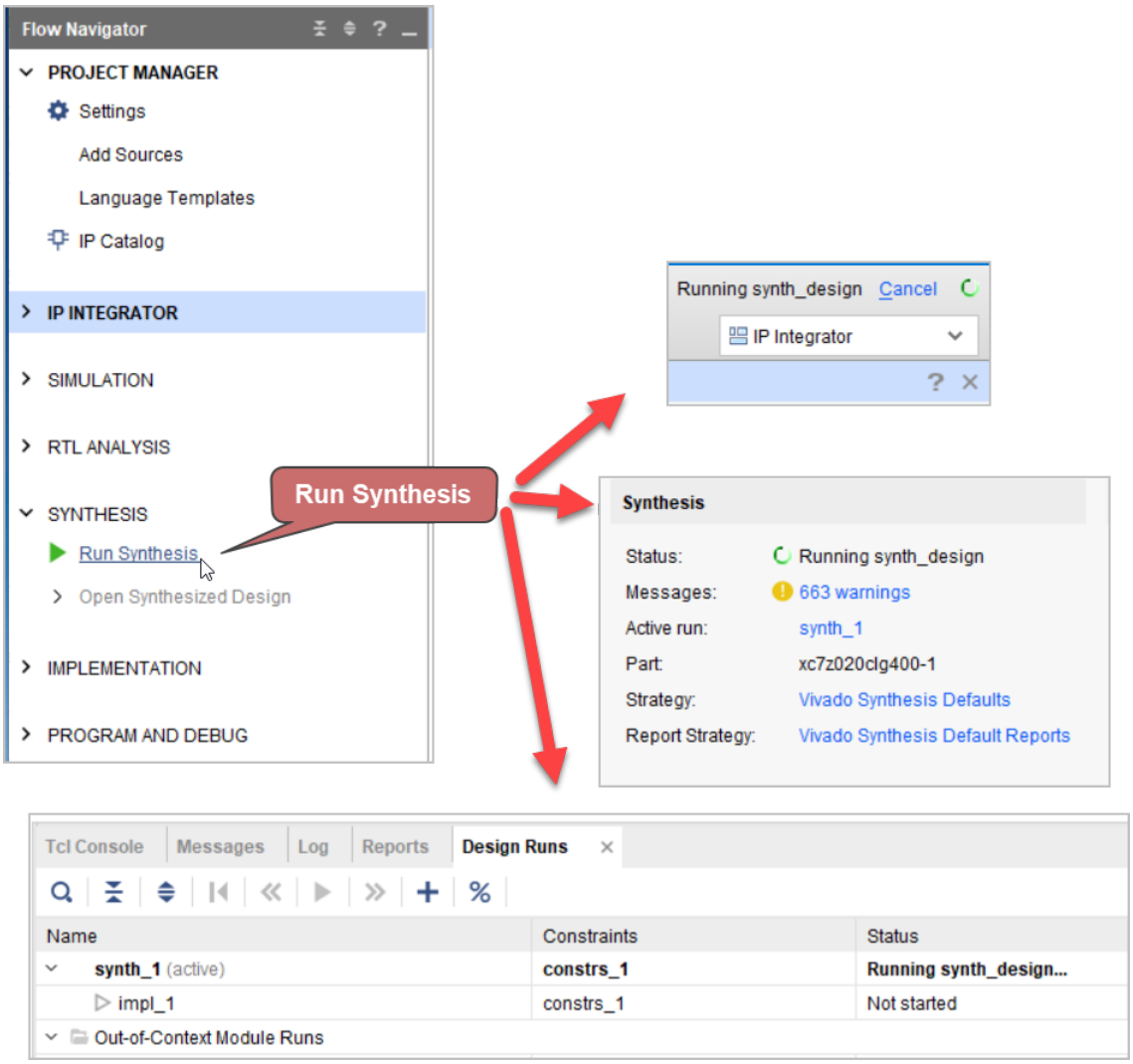


Figure 77. Run design synthesis.

2.4.2 Implement the Design

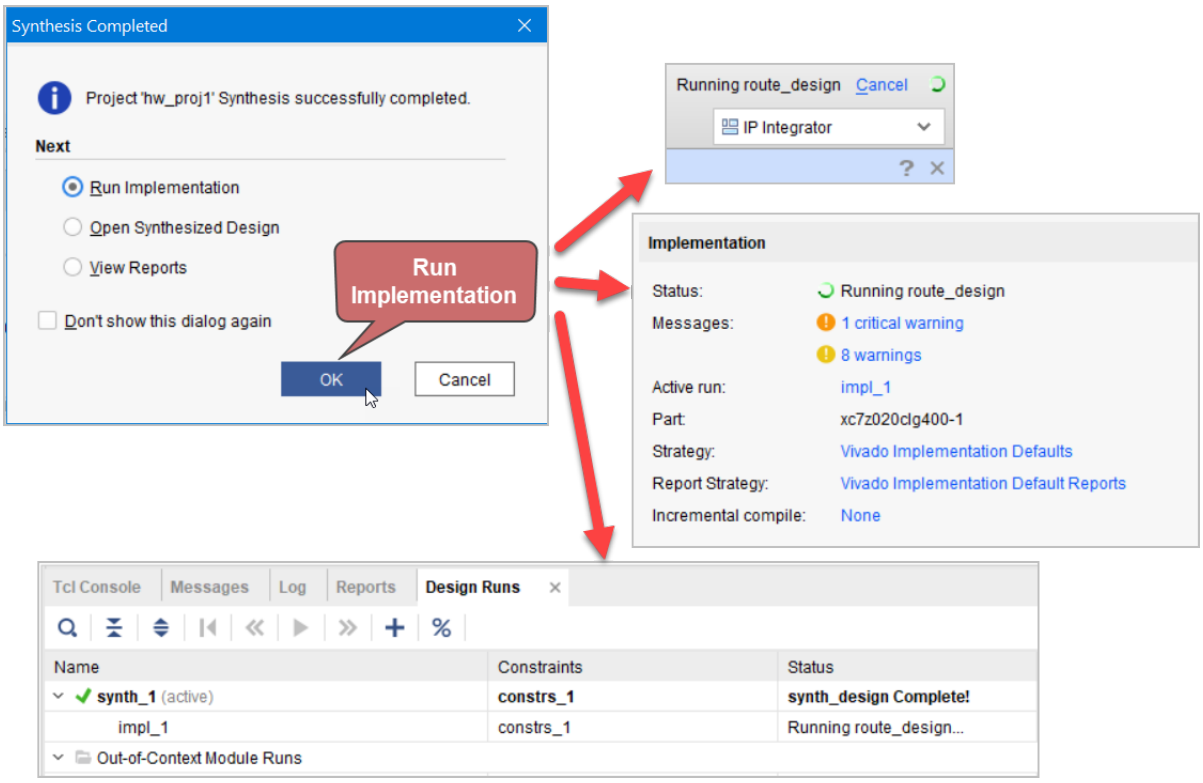


Figure 78. Implement the design.

2.4.3 Generate the Bitstream

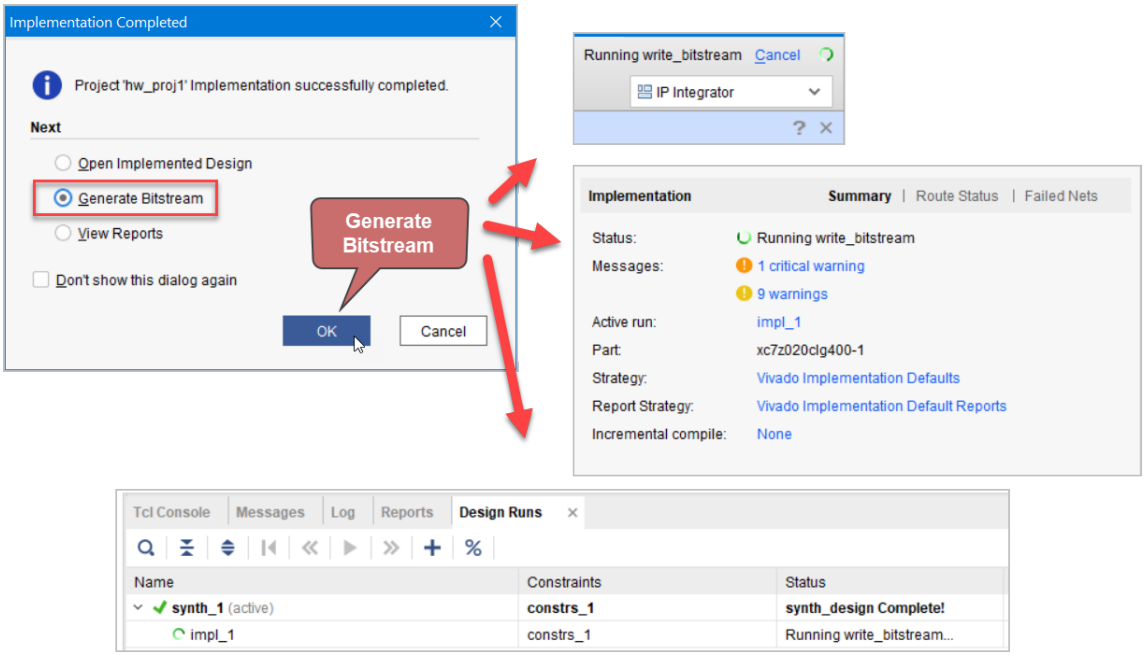


Figure 79. Generate the bitstream.

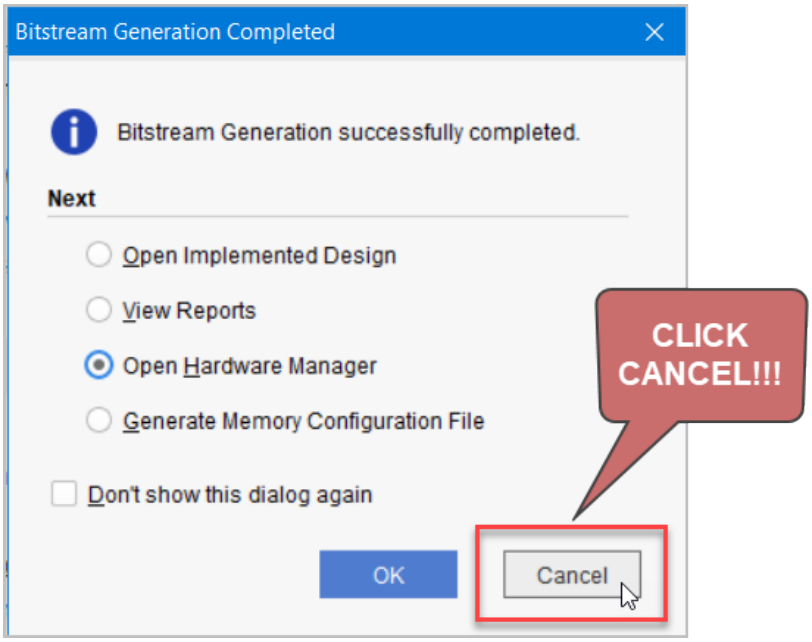


Figure 80. When the bitstream is generated, click Cancel (unless the user wants to open the HW Manager).

2.5 Hardware Hand-off

2.5.1 Export the Design

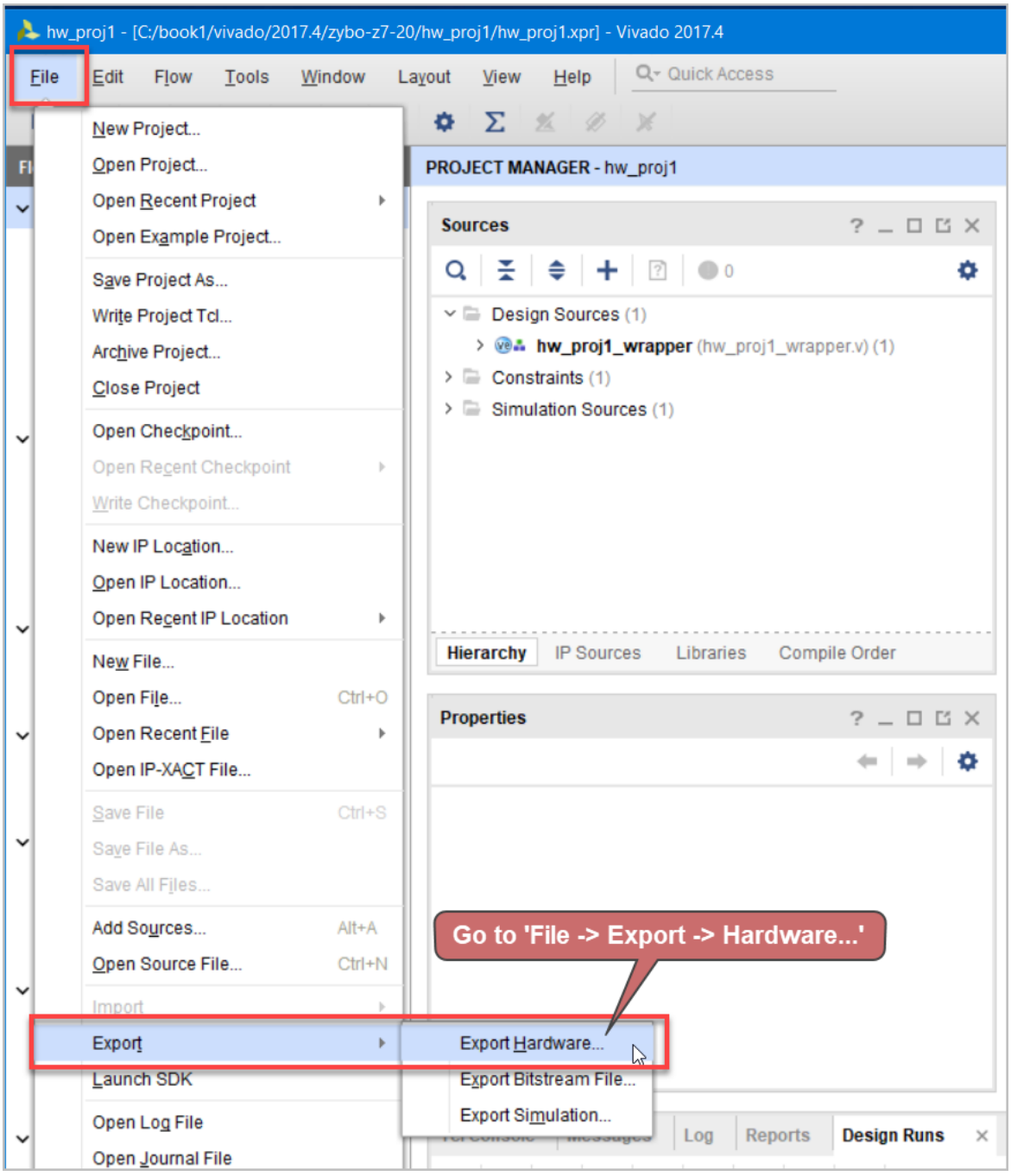


Figure 81. Export the design

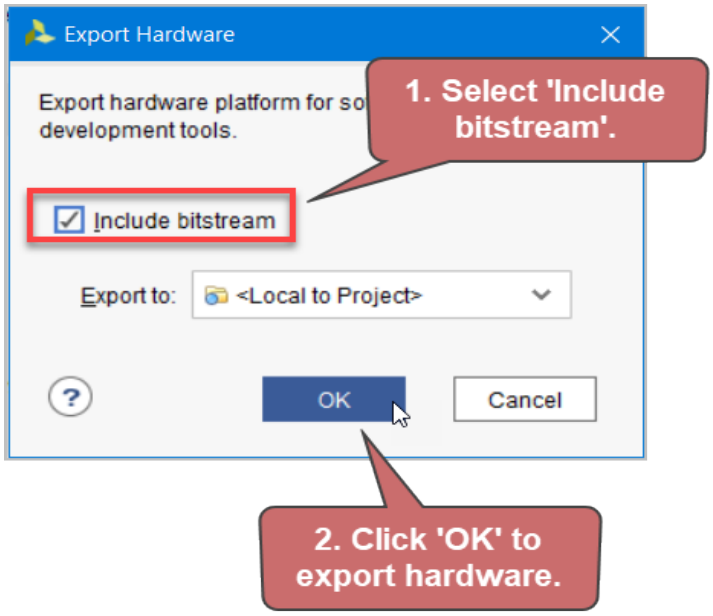


Figure 82. Include the bitstream and click OK.

2.5.2 Launch SDK

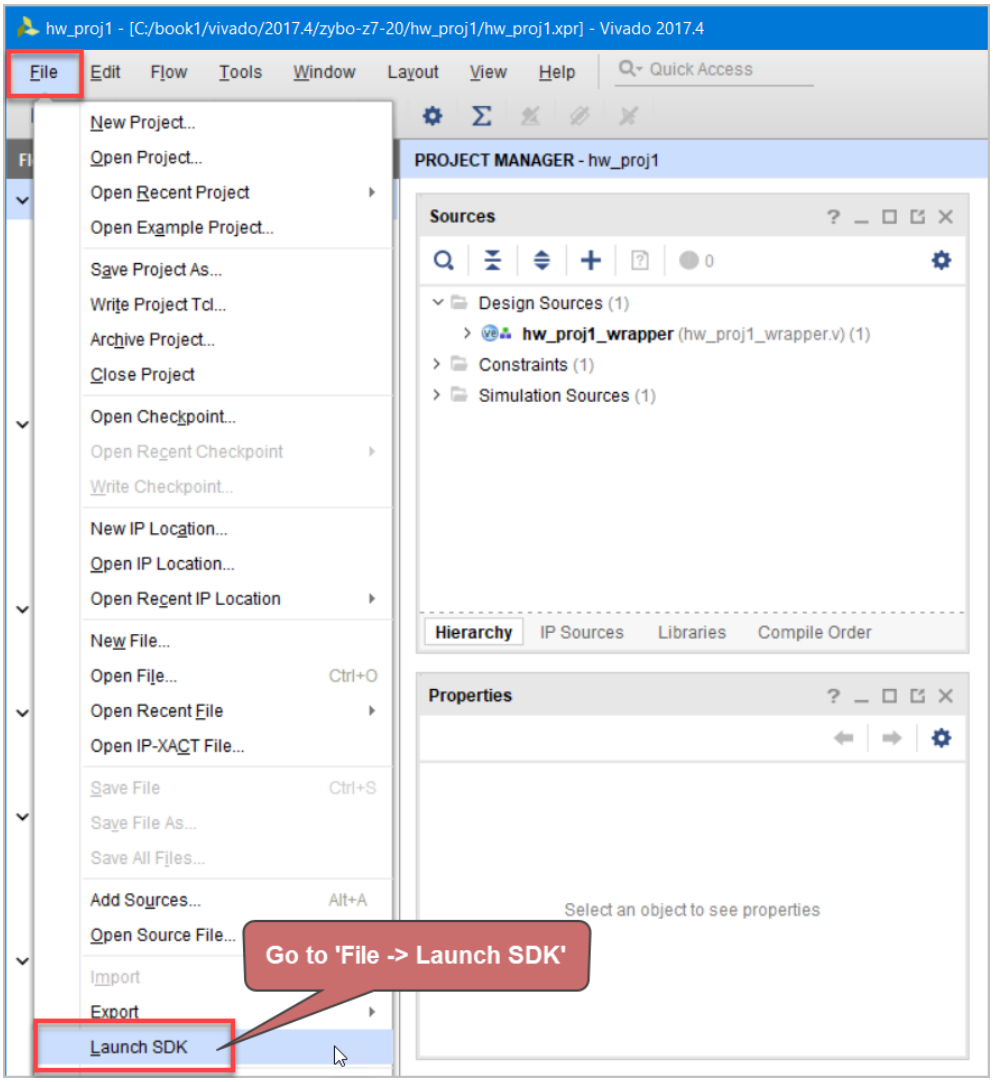


Figure 83. Launch SDK.

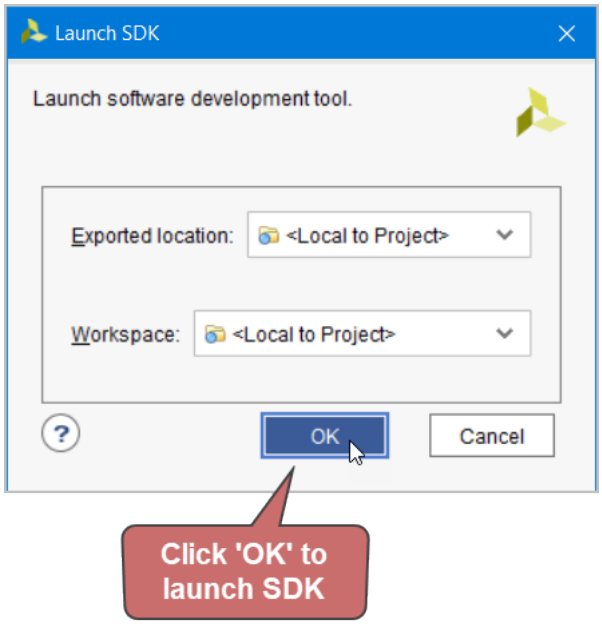


Figure 84. Click OK to launch SDK.

3 Software Development in Xilinx SDK

Figure 85. The Eclipse SDK opens after a short time (the user should not interfere during this time!). The hardware description file is automatically extracted in to the workspace.

3.1 Software Project 1: Hello World

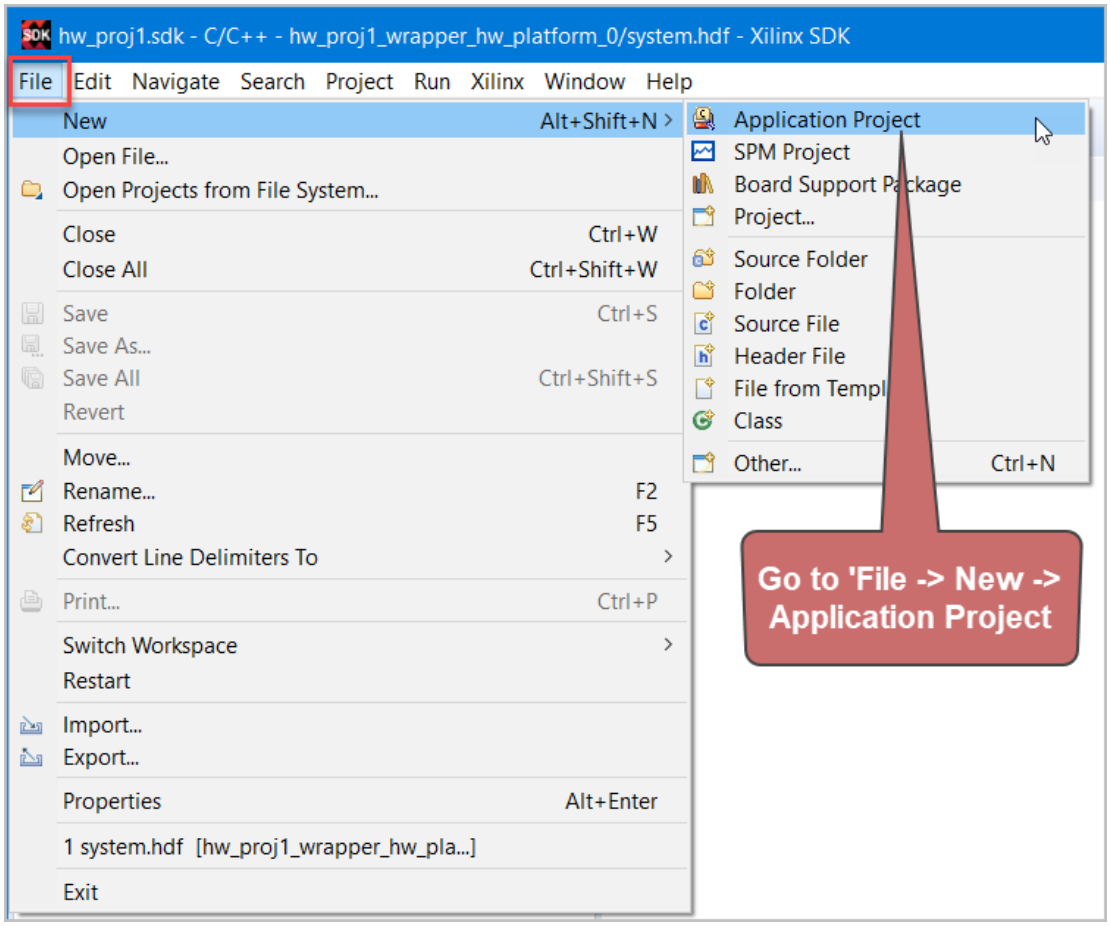


Figure 86. Go to File -> New -> Application Project

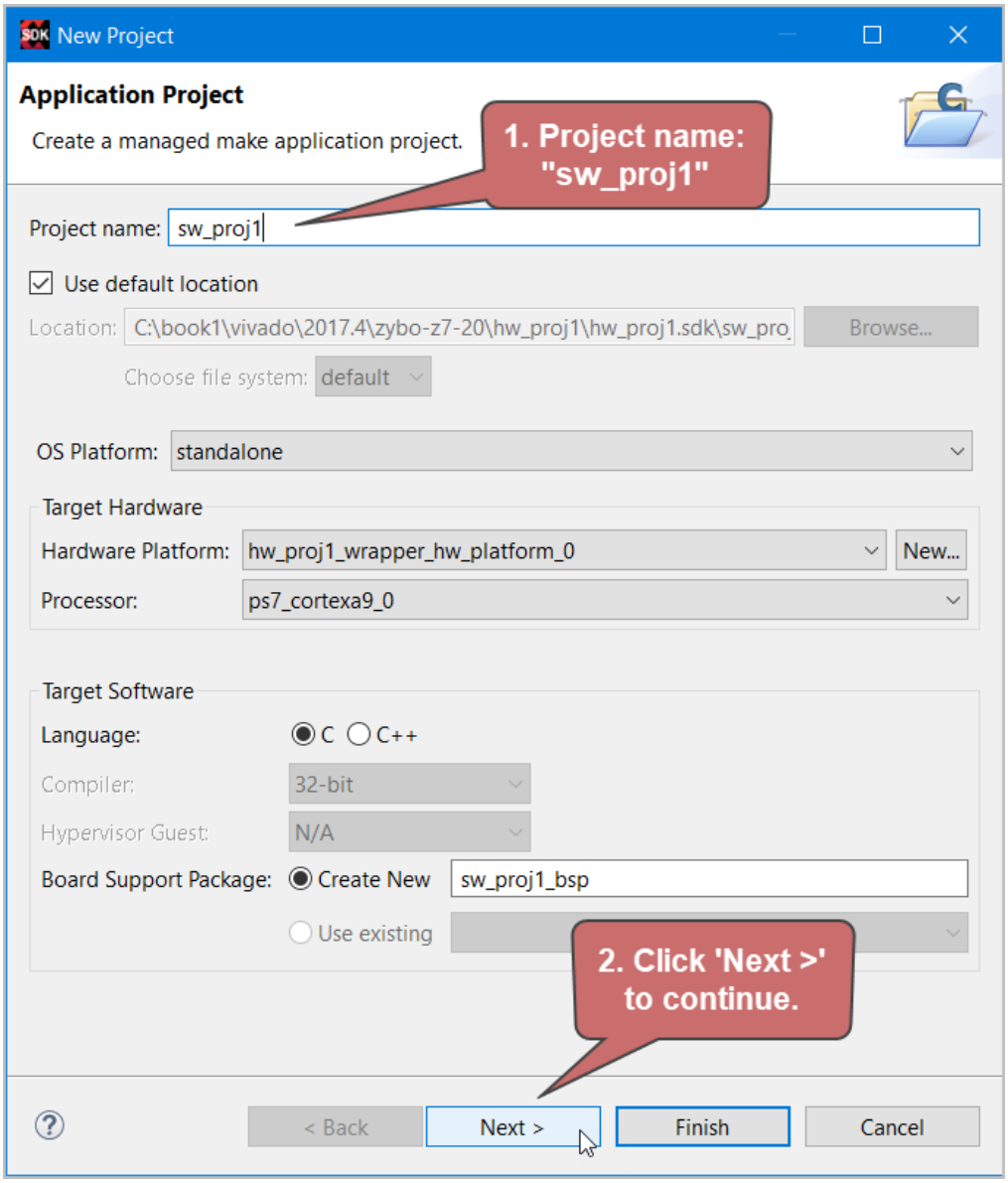


Figure 87. Call the project "sw_proj1" and click Next >.

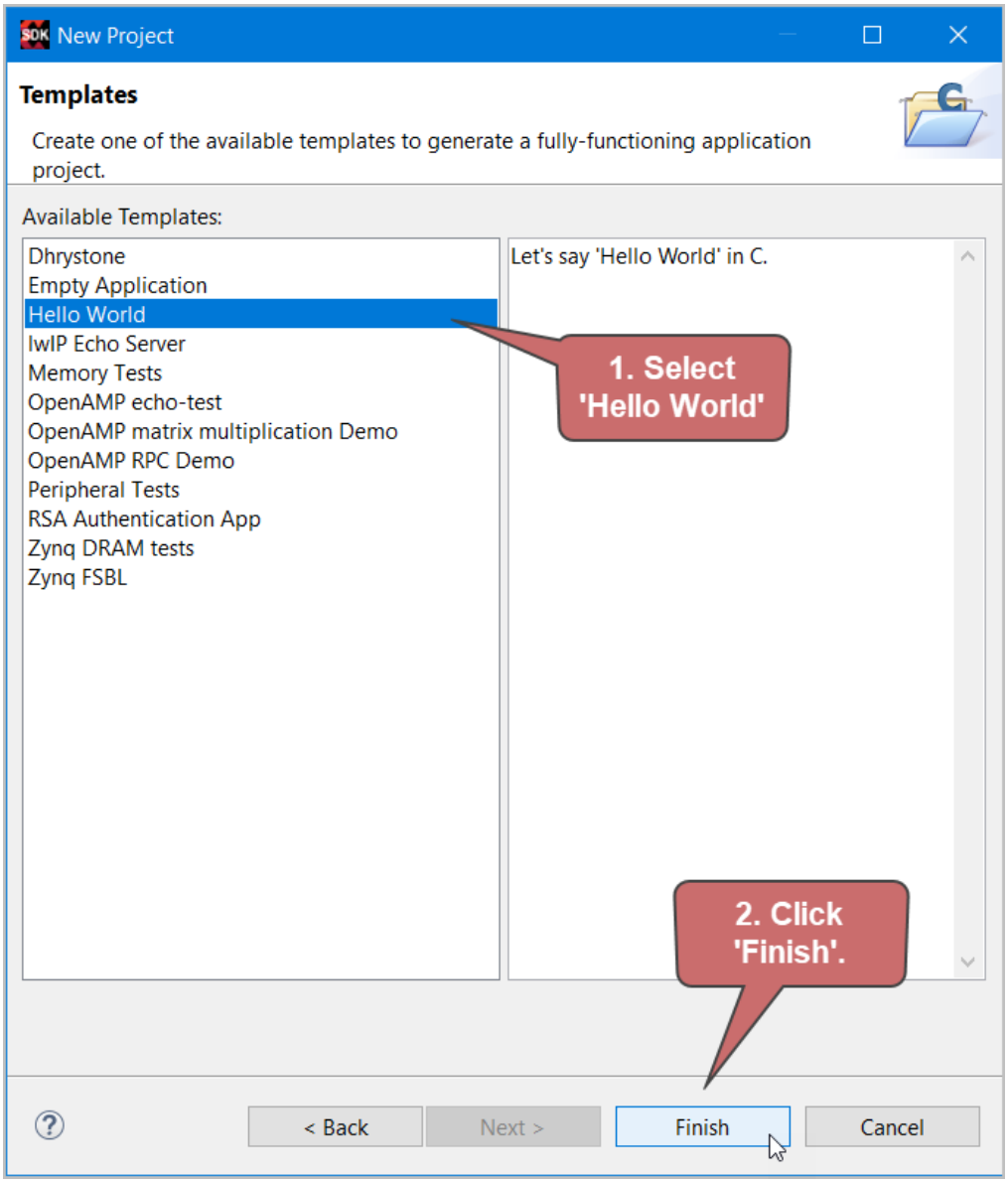


Figure 88. Ensure the Hello World template is selected, and click Finish.

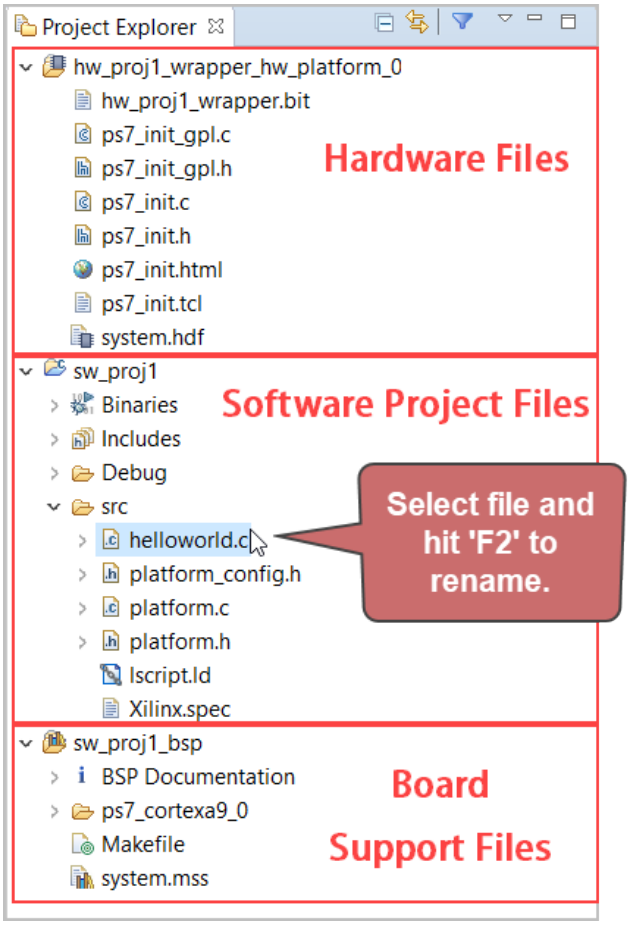


Figure 89. Rename the main source file

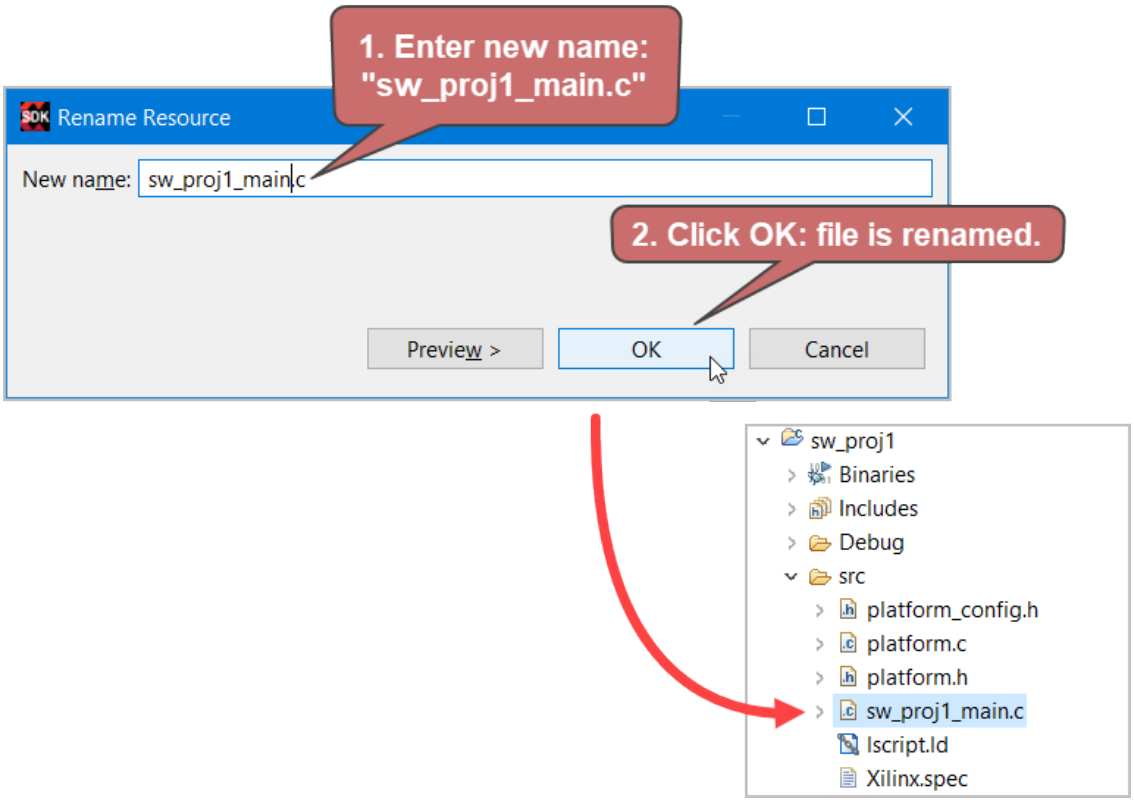


Figure 90. Change the project name to "sw_proj1_main.c"

```c
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"


int main()
{
    init_platform();

    print("Hello World\n\r");

    cleanup_platform();
    return 0;
}
```

3.1.1 Prepare the Platform

- On the Zybo-Z7-20, connect a Micro-B USB cable to J12.
- Power up the platform using SW4.

3.1.2 Open Console Application

Here, Tera Term is used, although the console in the XSDK could also be used (or indeed any other suitable terminal program). Suitable settings are shown in the following figures. The terminal and serial port settings are important, but the window and font settings are optional.

Note that in this particular case, the platform is found on COM Port 6, but this will vary from system to system.

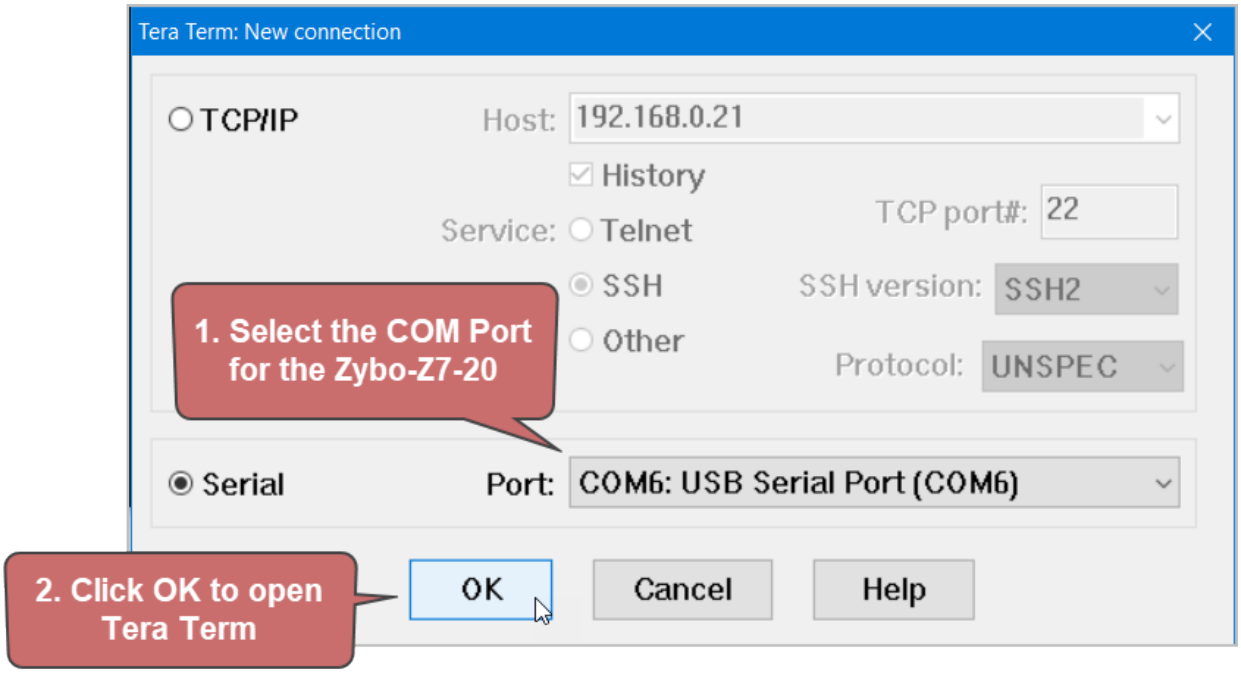


Figure 91. Open the connection to the platform.

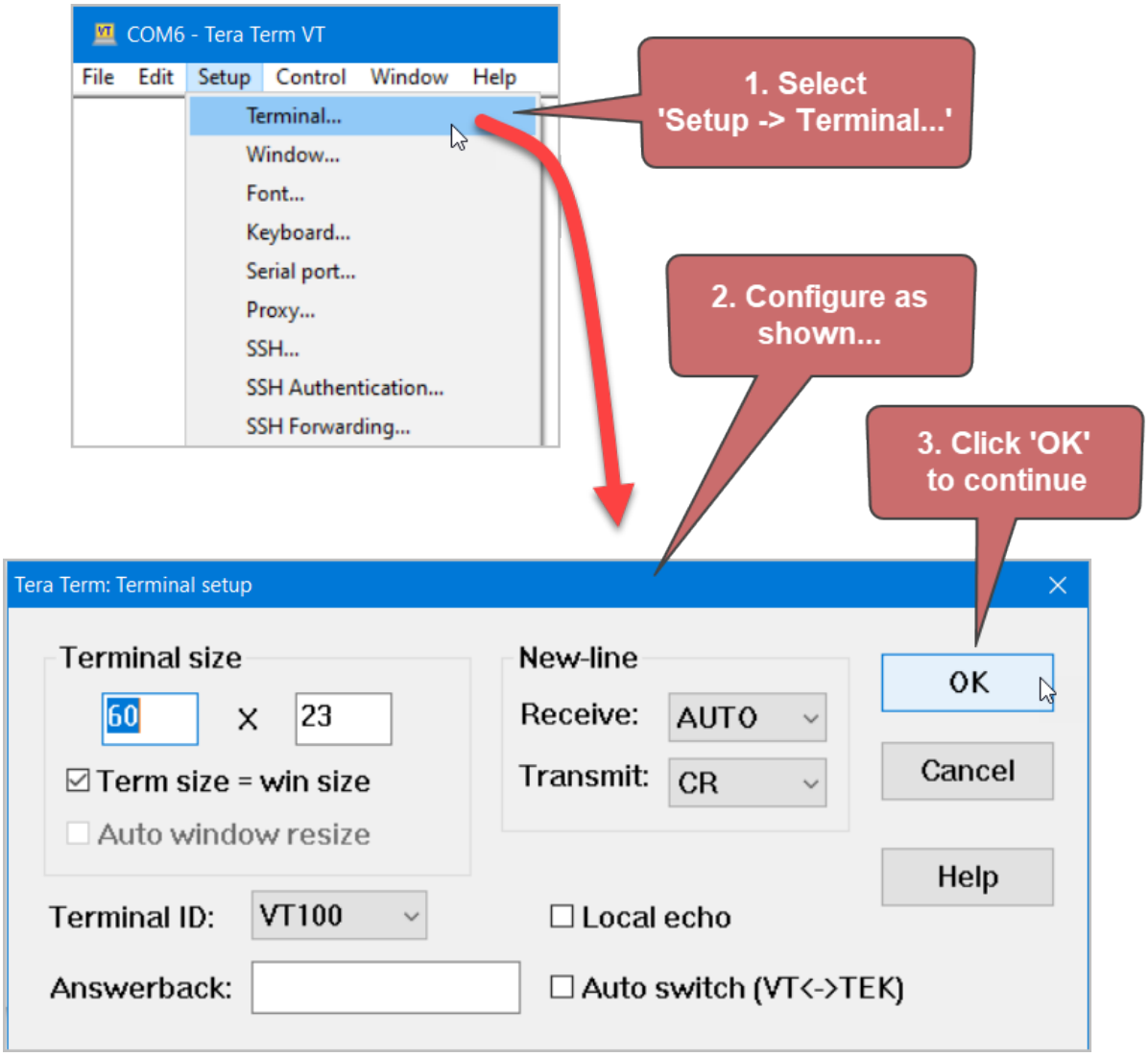


Figure 92. Configure the terminal. The New-line settings in particular are important.

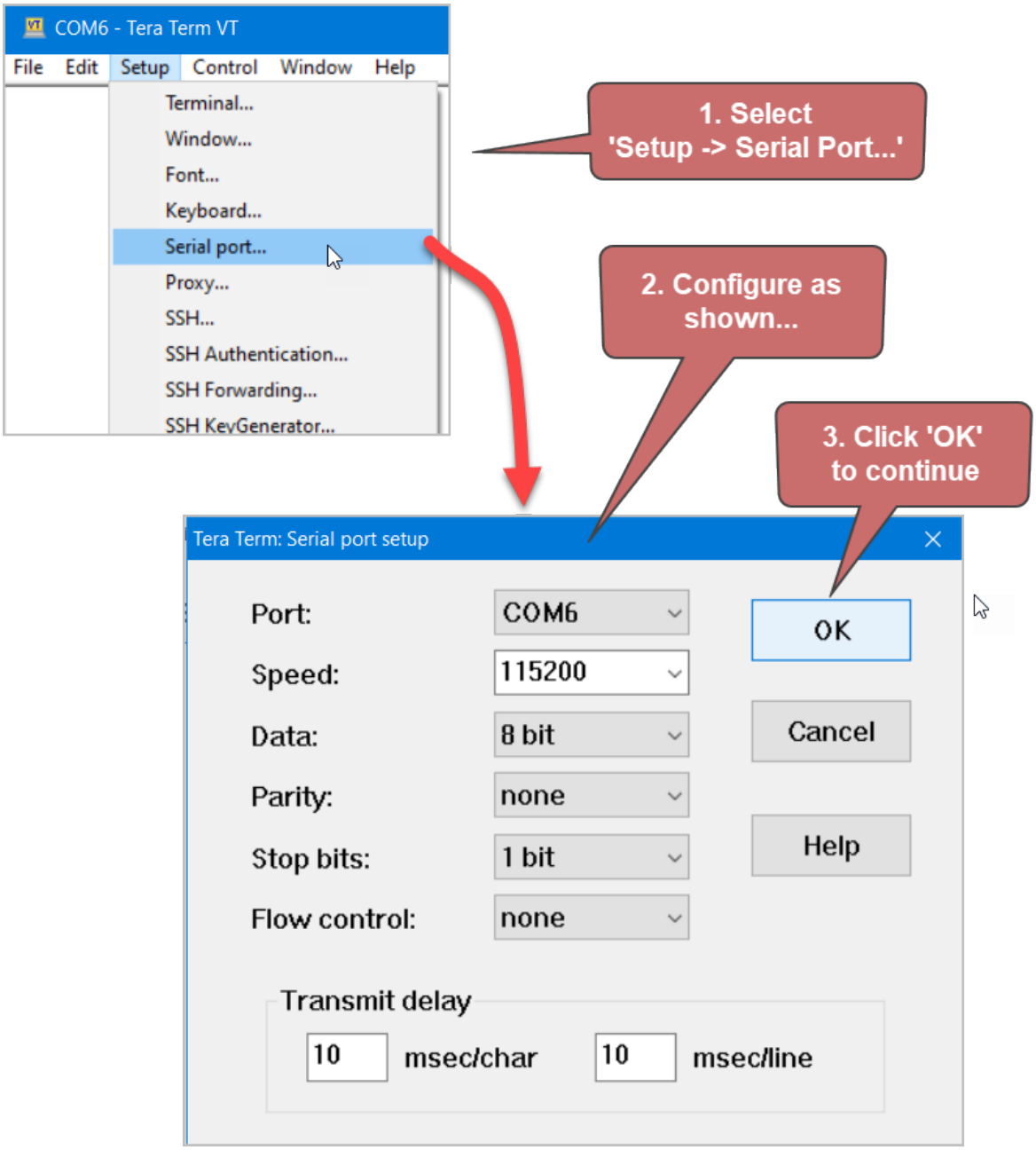


Figure 93. Configure the serial port settings.

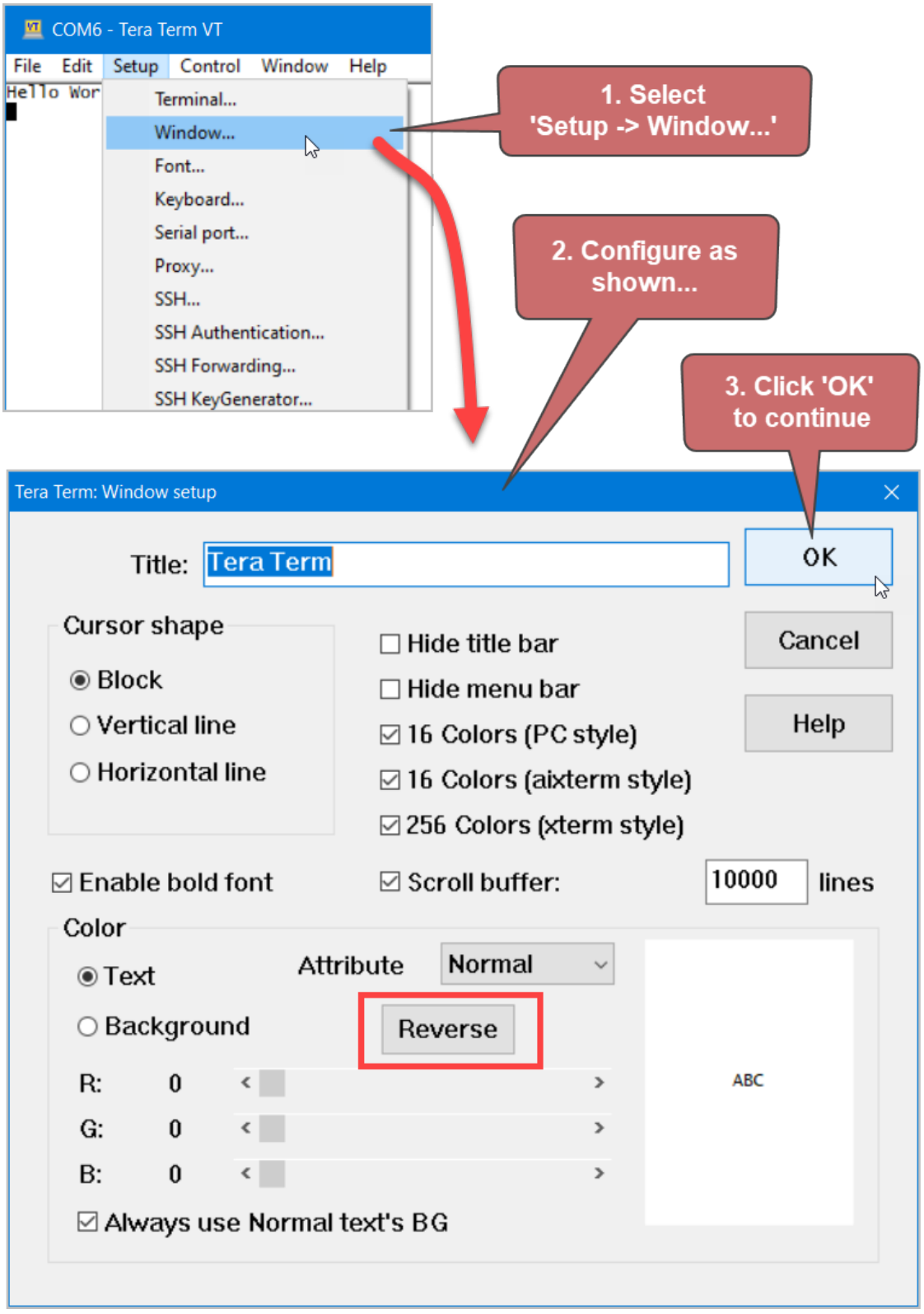


Figure 94. Optionally, reconfigure the window with a white background and black text.

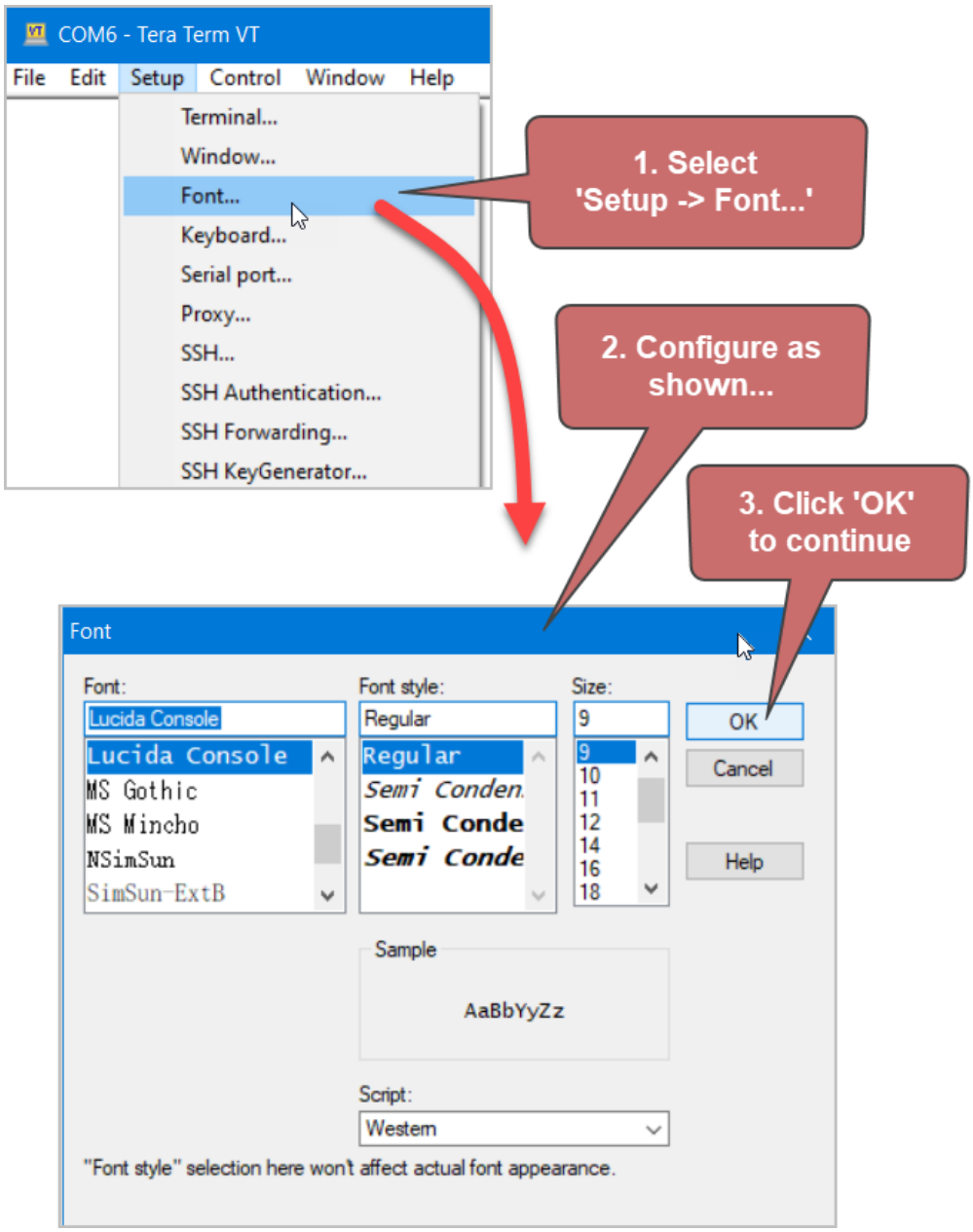


Figure 95. Optionally, change the font.

3.1.3 Run the Program

First, open the XSCT console. This is not a necessary step, but it does provide useful information on what happens when the program is launched.

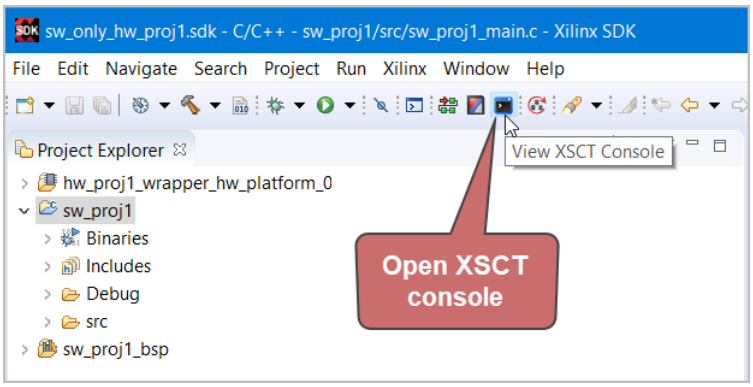


Figure 96. Open the XSCT console

To run the program, the first step is to create a Run configuration. Right-click on the project, and select 'Run As -> Run Configurations...' option.

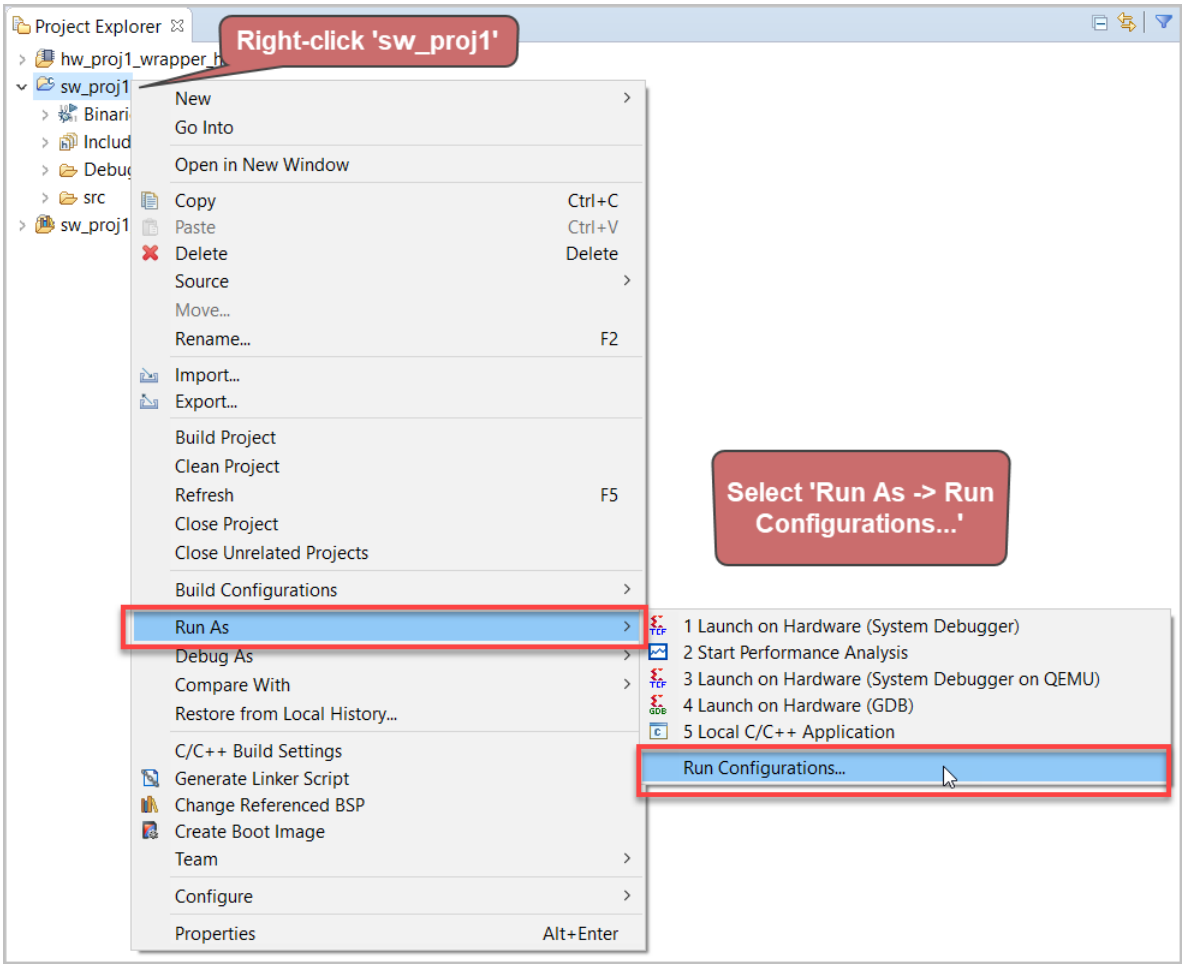


Figure 97. Right-click the software project to create a Run Configuration

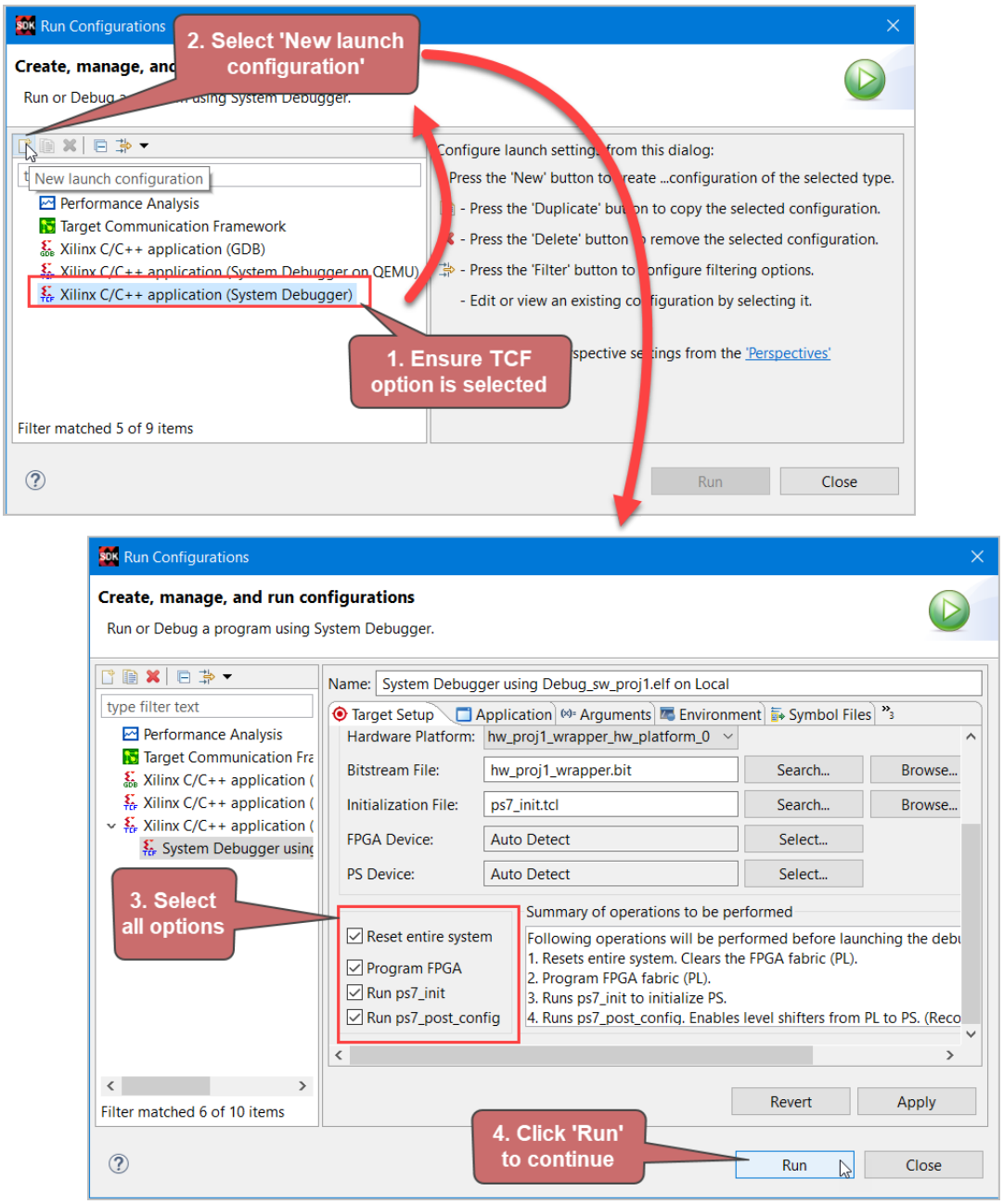


Figure 98. Create a TCF (i.e. System Debugger) configuration, and execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

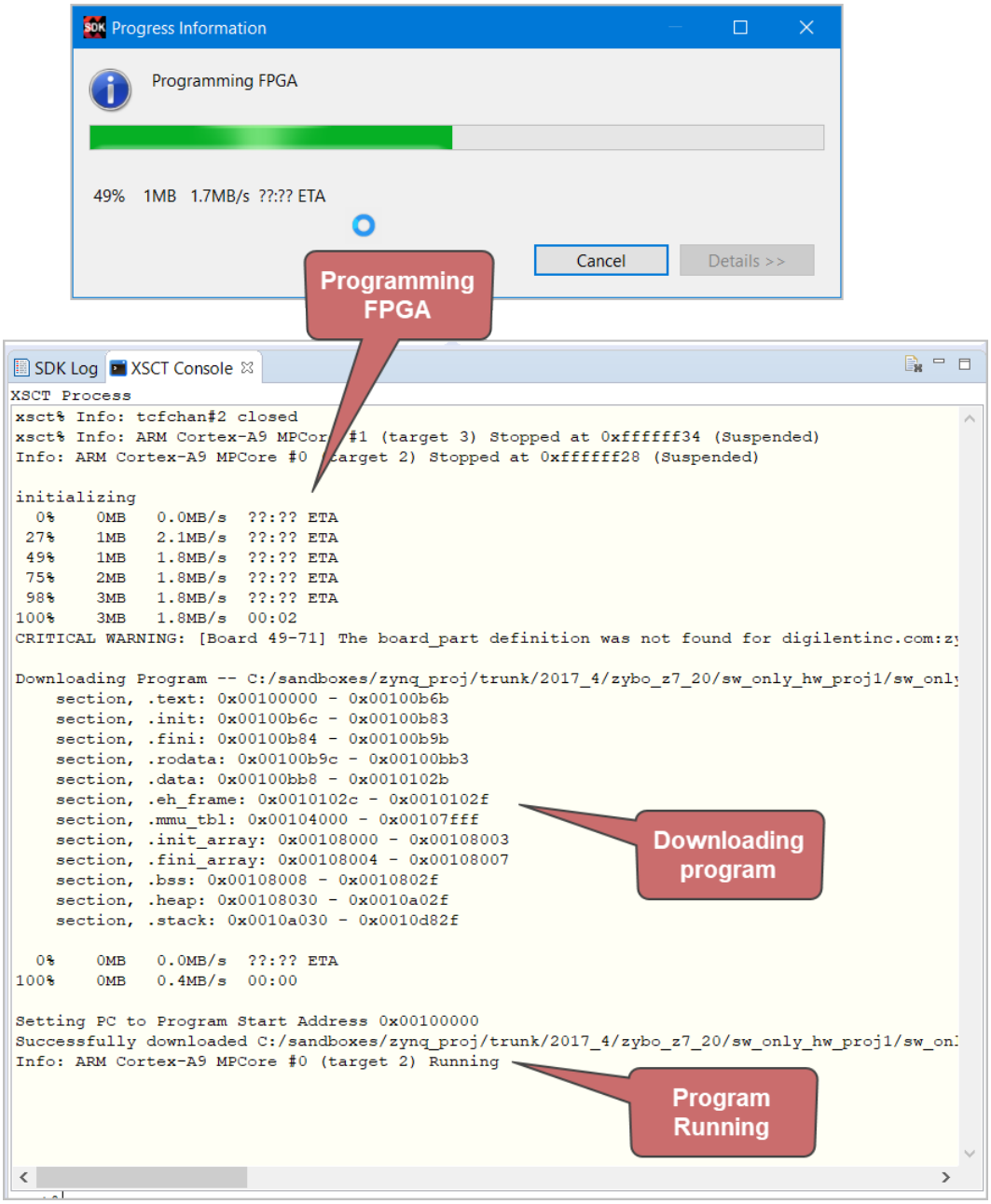


Figure 99. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

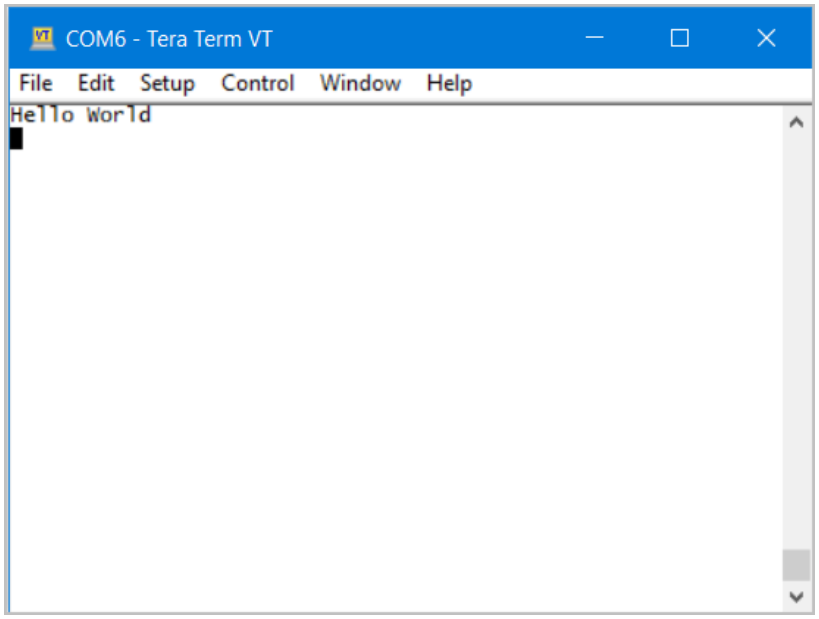


Figure 100. The program output is displayed in the console.

The “Hello World” message is displayed in the console.

3.2 Software Project 2: Zynq GPIO

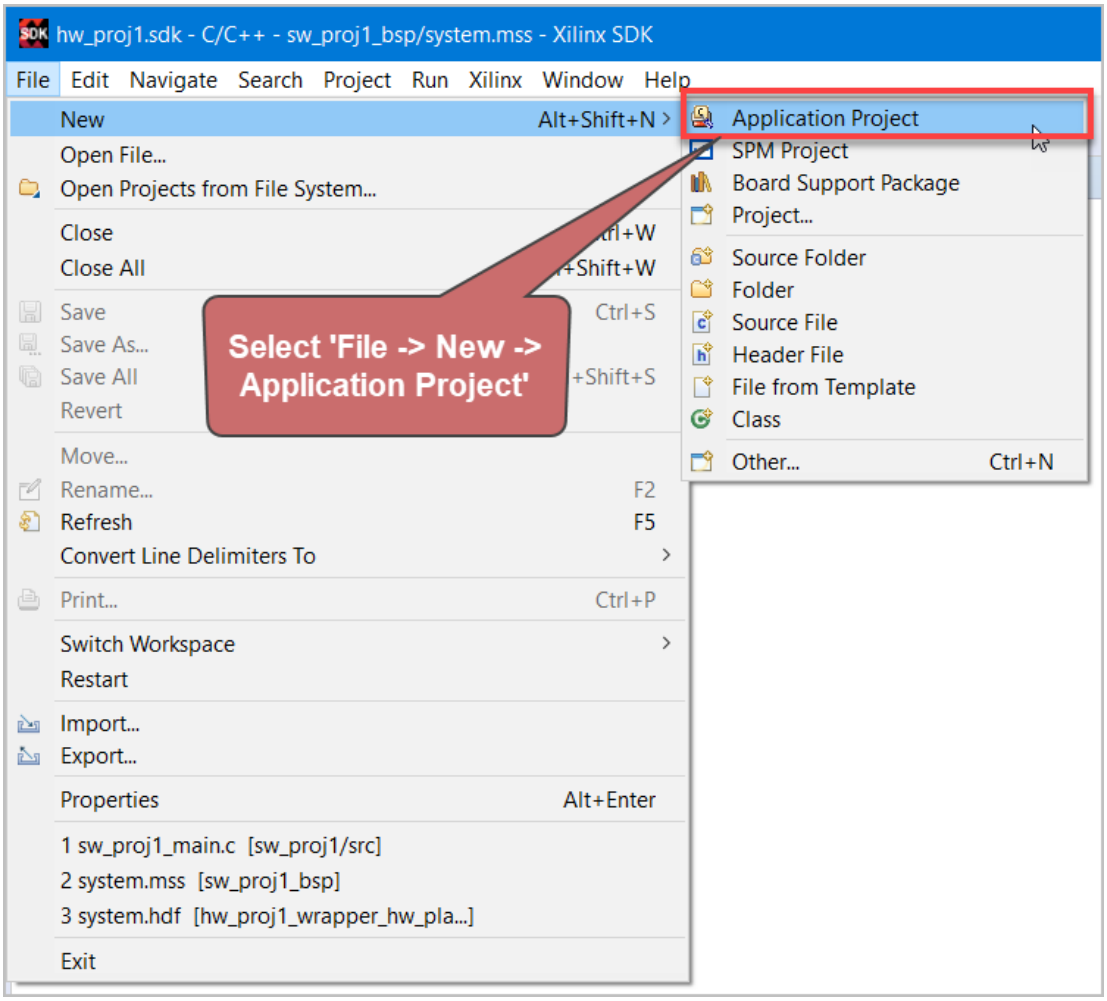


Figure 101. Select File->New-> Application Project

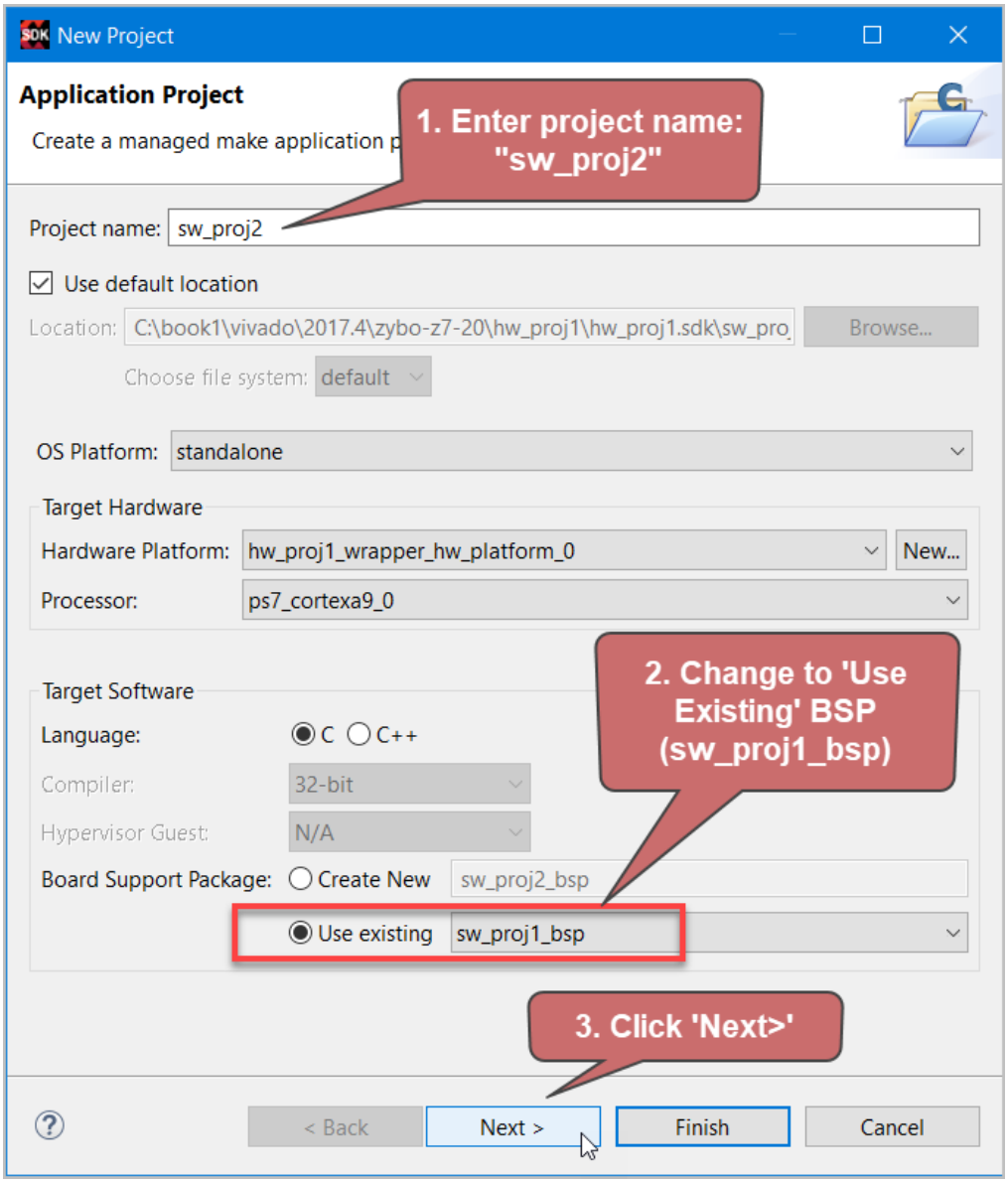


Figure 102. Enter the project details (part 1)

5. Enter the project name: sw_proj2
6. Board Support Package: Use existing
7. Click Next (don't click Finish!)

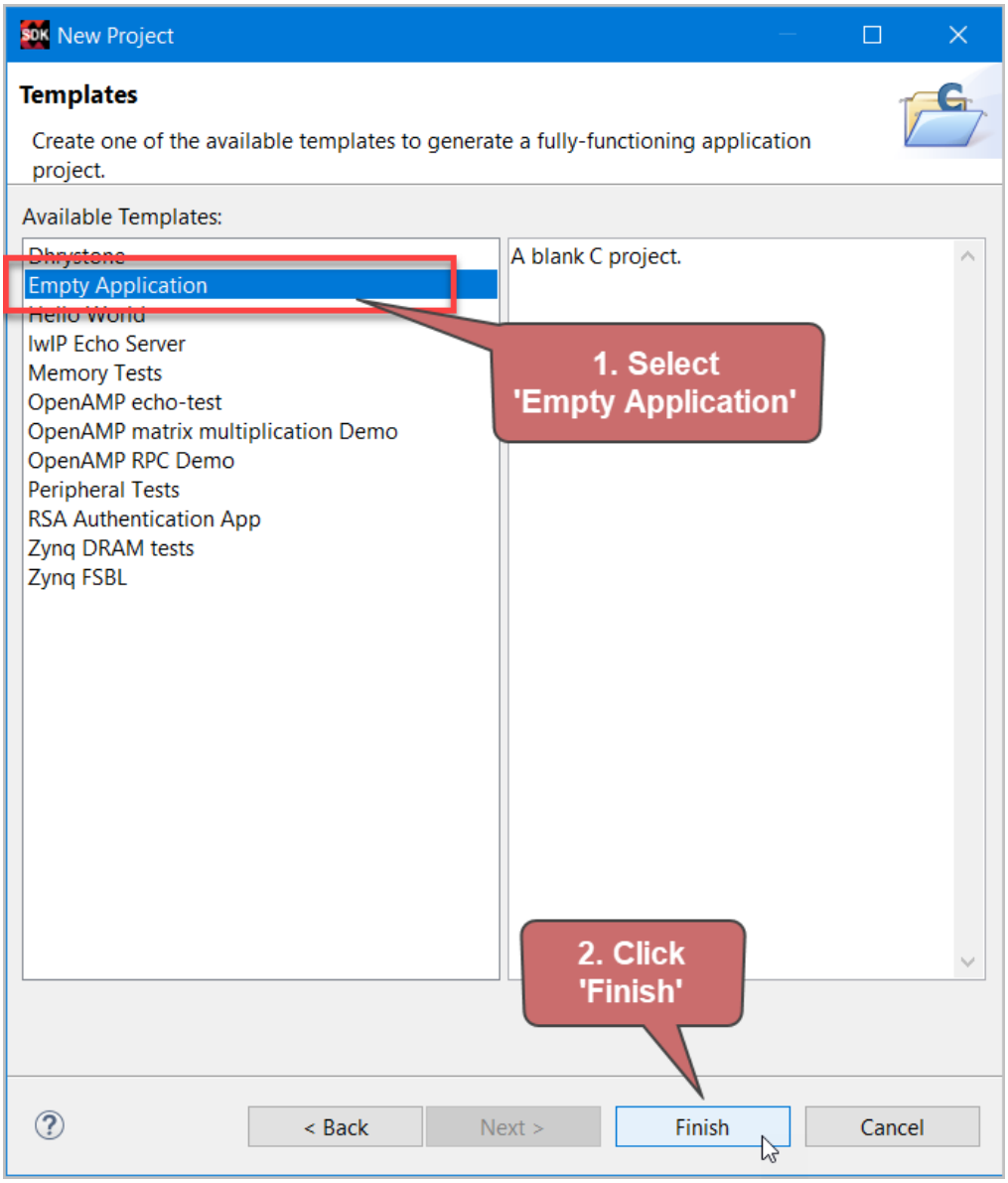


Figure 103. Enter the project details (part 2)

1. Select Empty Application
2. Click Finish

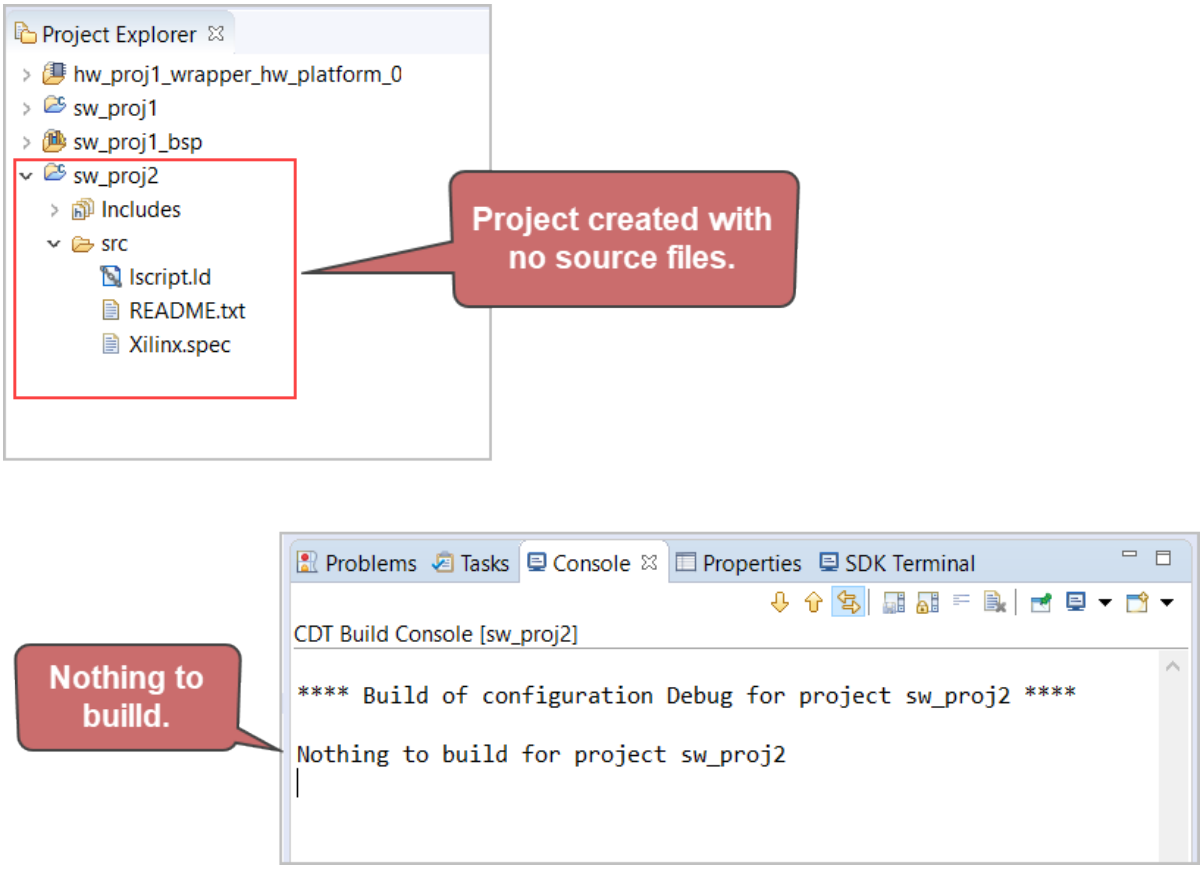


Figure 105. Project created with no source files

A project named "sw_proj2" is created with no project files. The CDT Build Console confirms that there is nothing to build.

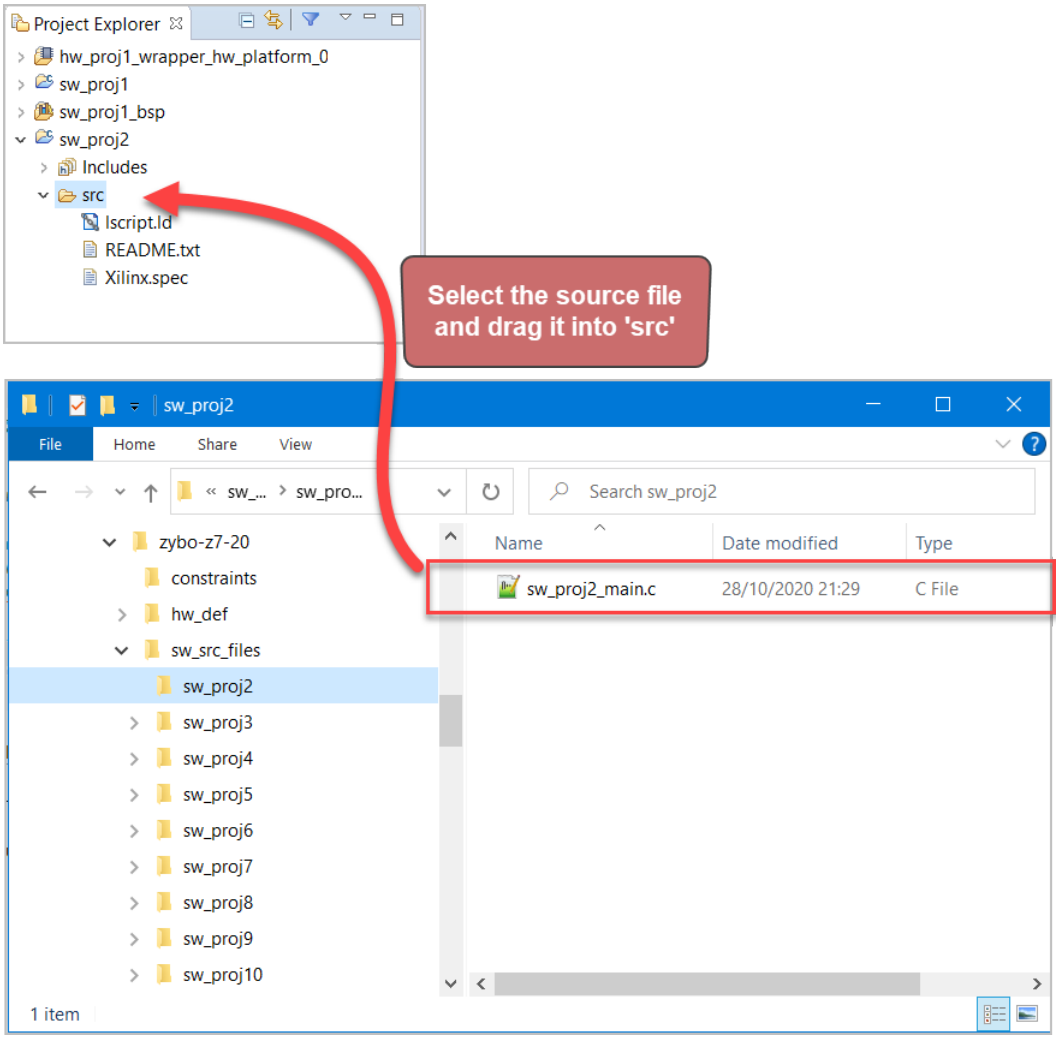


Figure 106. Drag the source file from Windows Explorer into the SDK project

Open the location where the source file for software project 2 is saved on your PC. Drag the file from Windows explorer directly into the "**src**" folder in SDK.

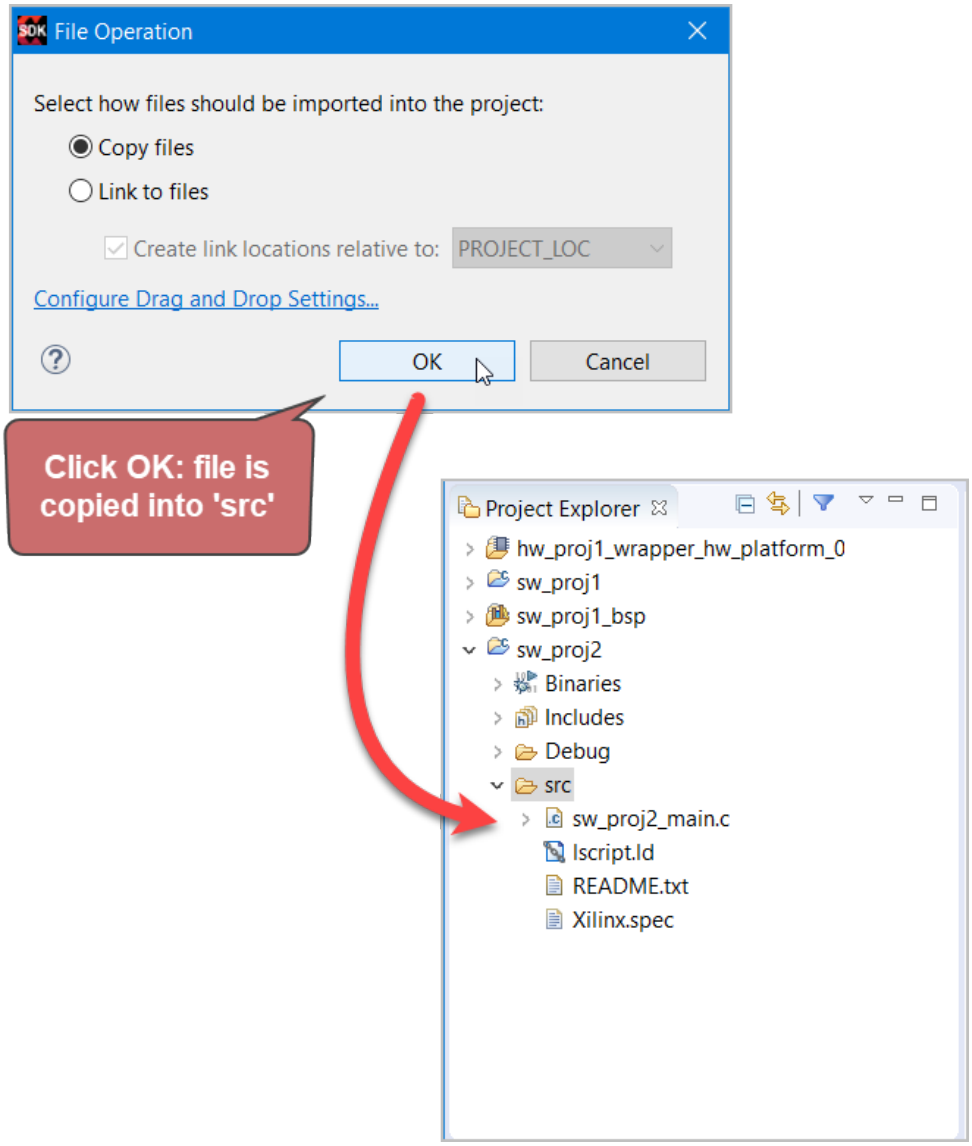


Figure 107. Click OK, and the file will be copied into the project.

In the File Operation dialog box, ensure “Copy files” is checked, and click OK.

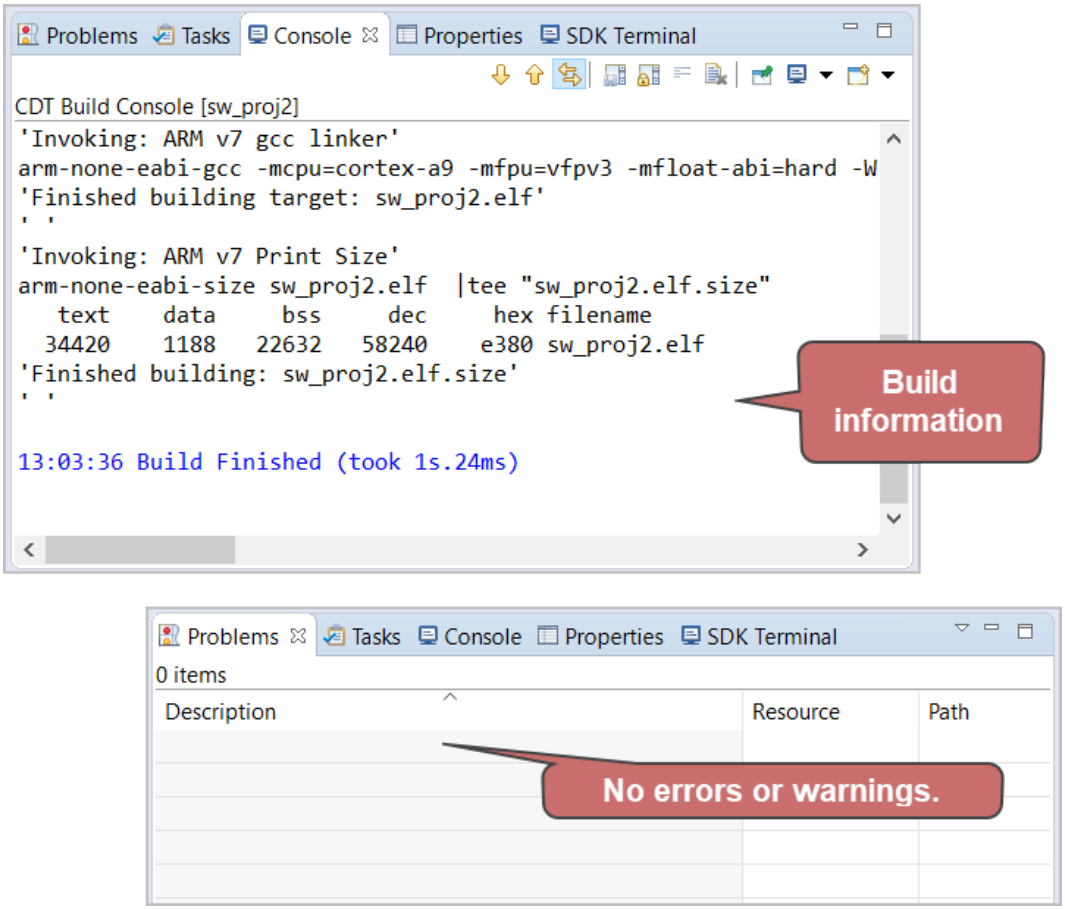


Figure 108. The project should build successfully

If "Build Automatically" is selected, the project should build successfully, with no errors or warnings.

[Exceptions: SDK 2018.3/SDK 2019.1 may indicate a warning as follows, which can be ignored:

#pragma message: For the sleep routines, Global timer is being used

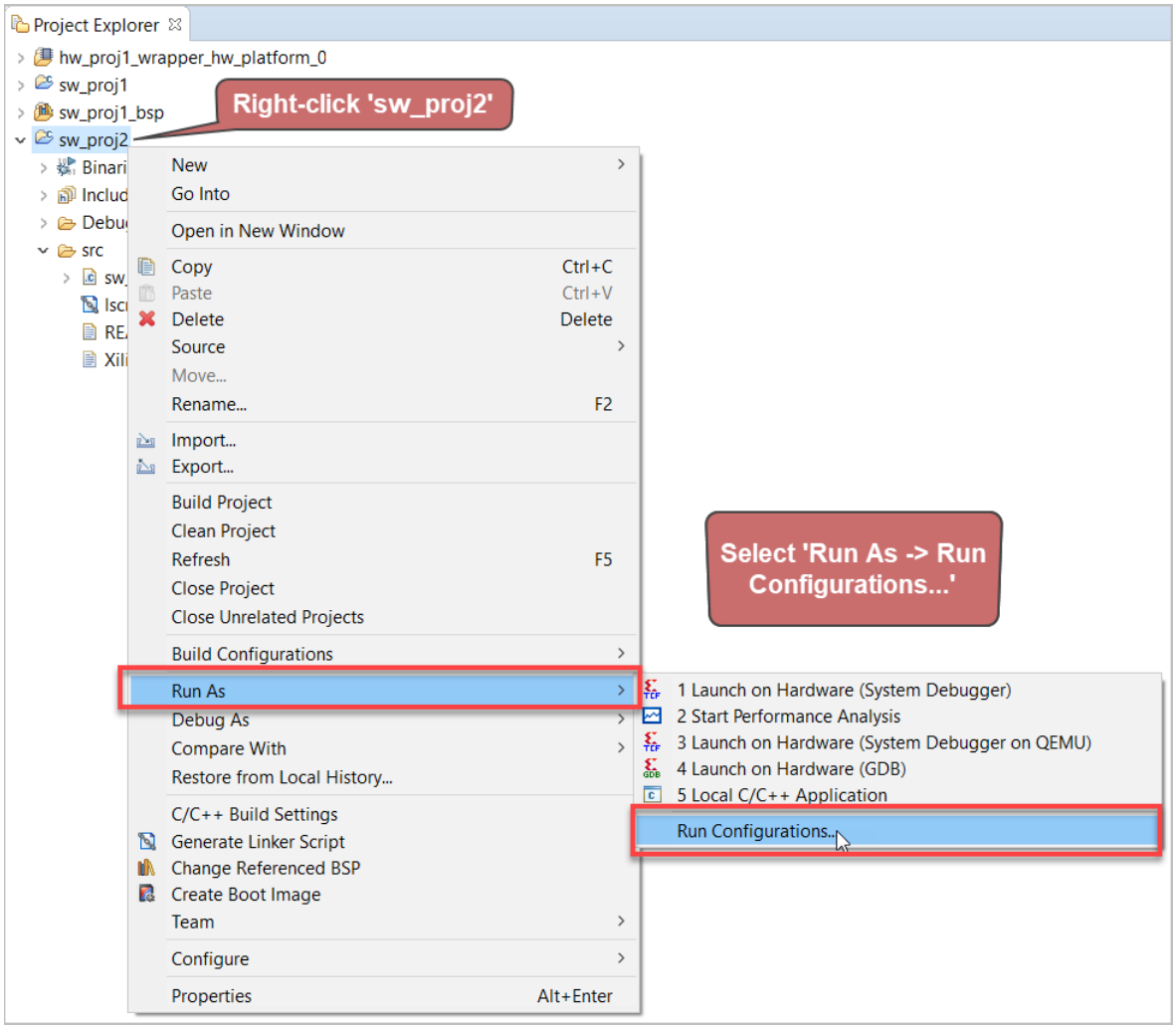


Figure 109. Right-click the software project and create a Run Configuration

To run the program, the first step is to create a Run configuration. Right-click on the project, and select the 'Run Configurations...' option.

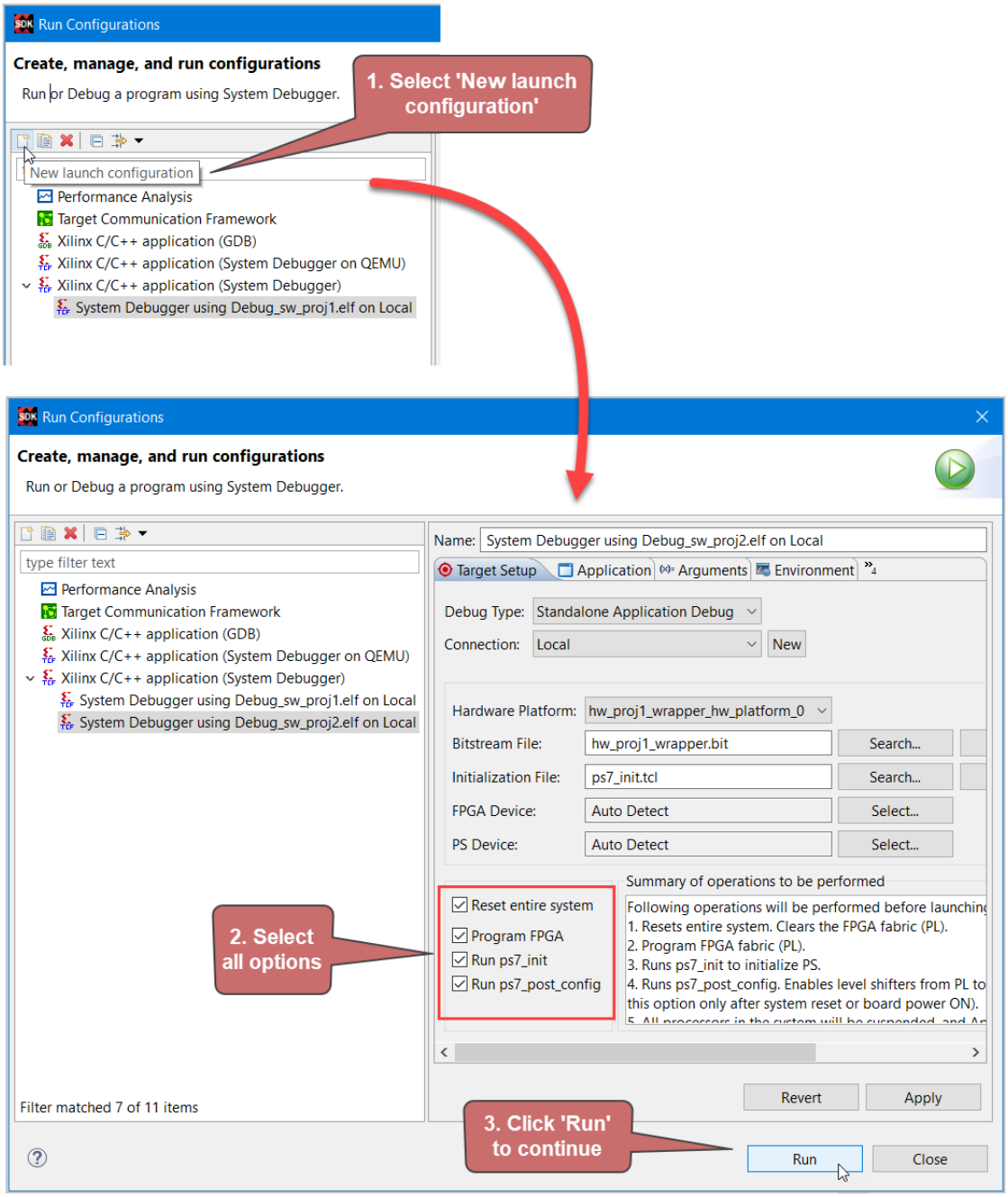


Figure 110. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

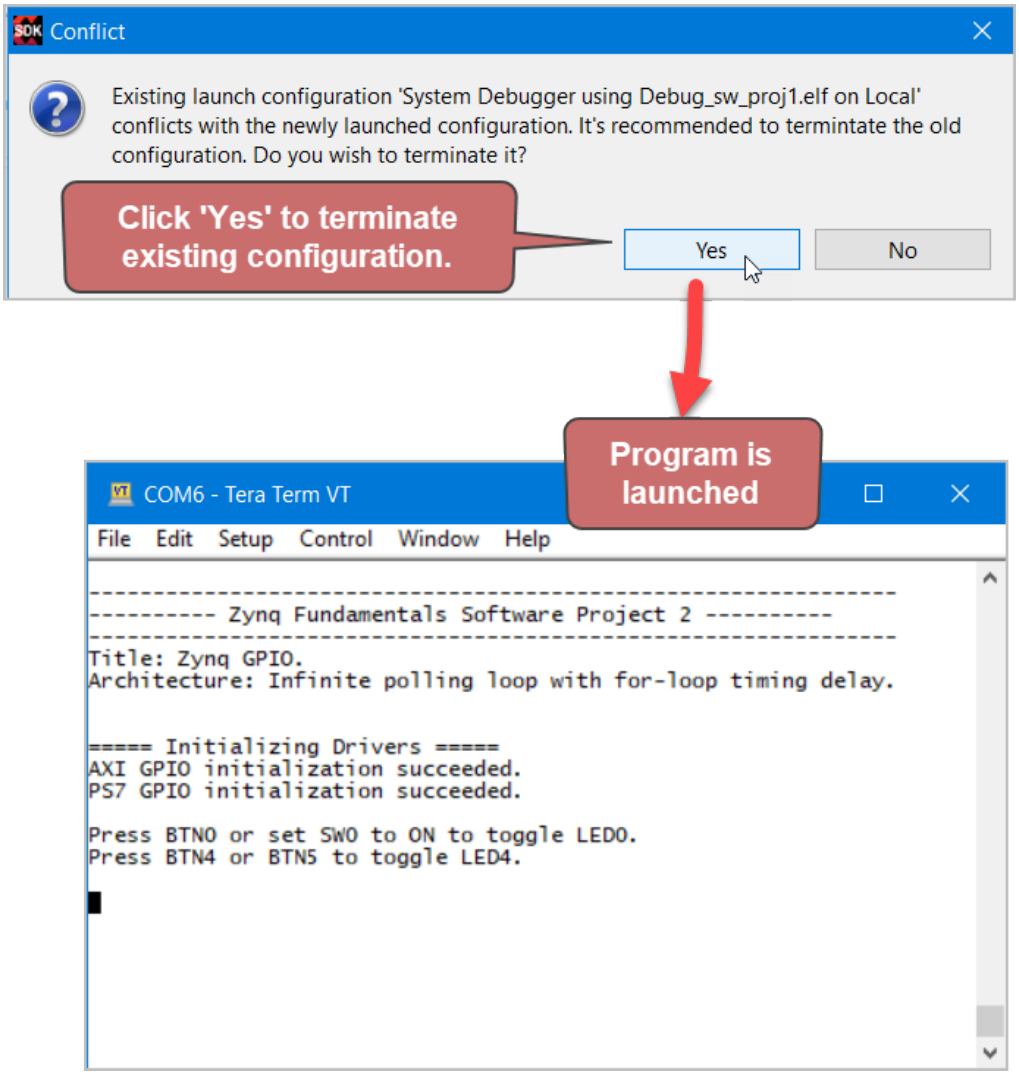


Figure 111. Terminal output for Software Project 2

If prompted, select 'Yes' to terminate any existing configuration. The FPGA will be programmed and the application will be downloaded to the processor. The terminal program should display the results for launching Software Project 2.

3.3 Software Project 3: Structured Program

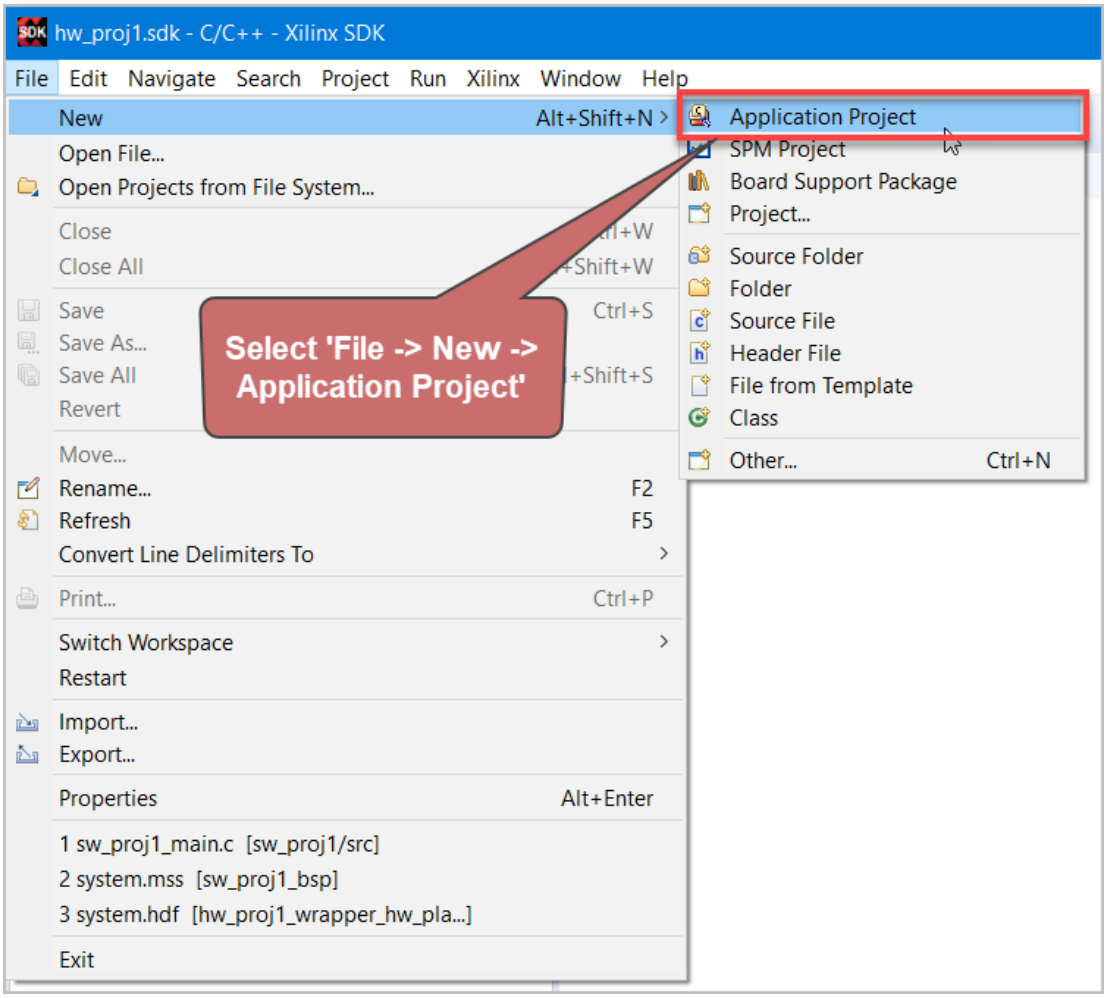


Figure 112. Select File->New-> Application Project

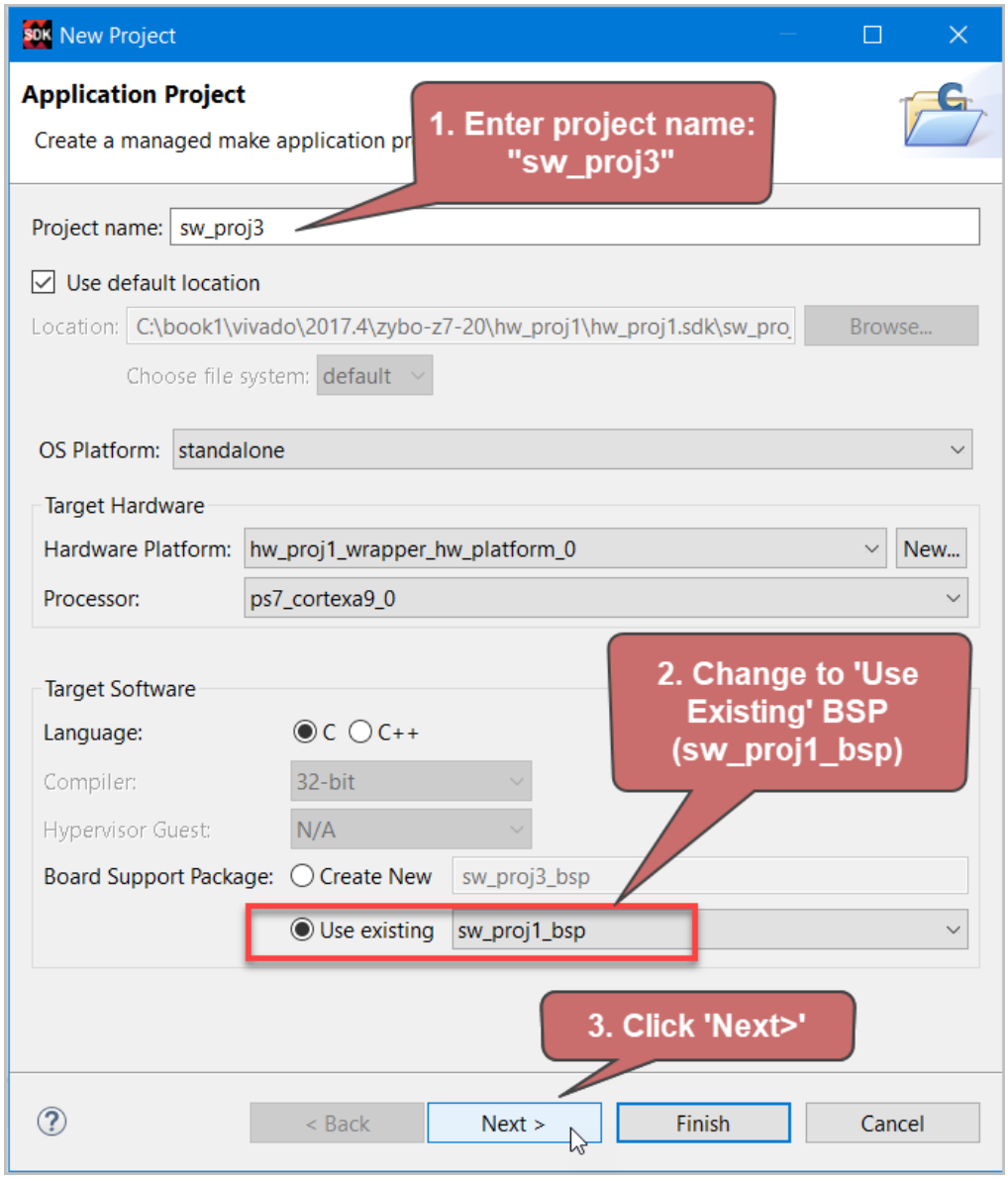


Figure 113. Enter the project details (part 1)

1. Enter the project name: sw_proj3
2. Board Support Package: Use existing
3. Click Next (don't click Finish!)

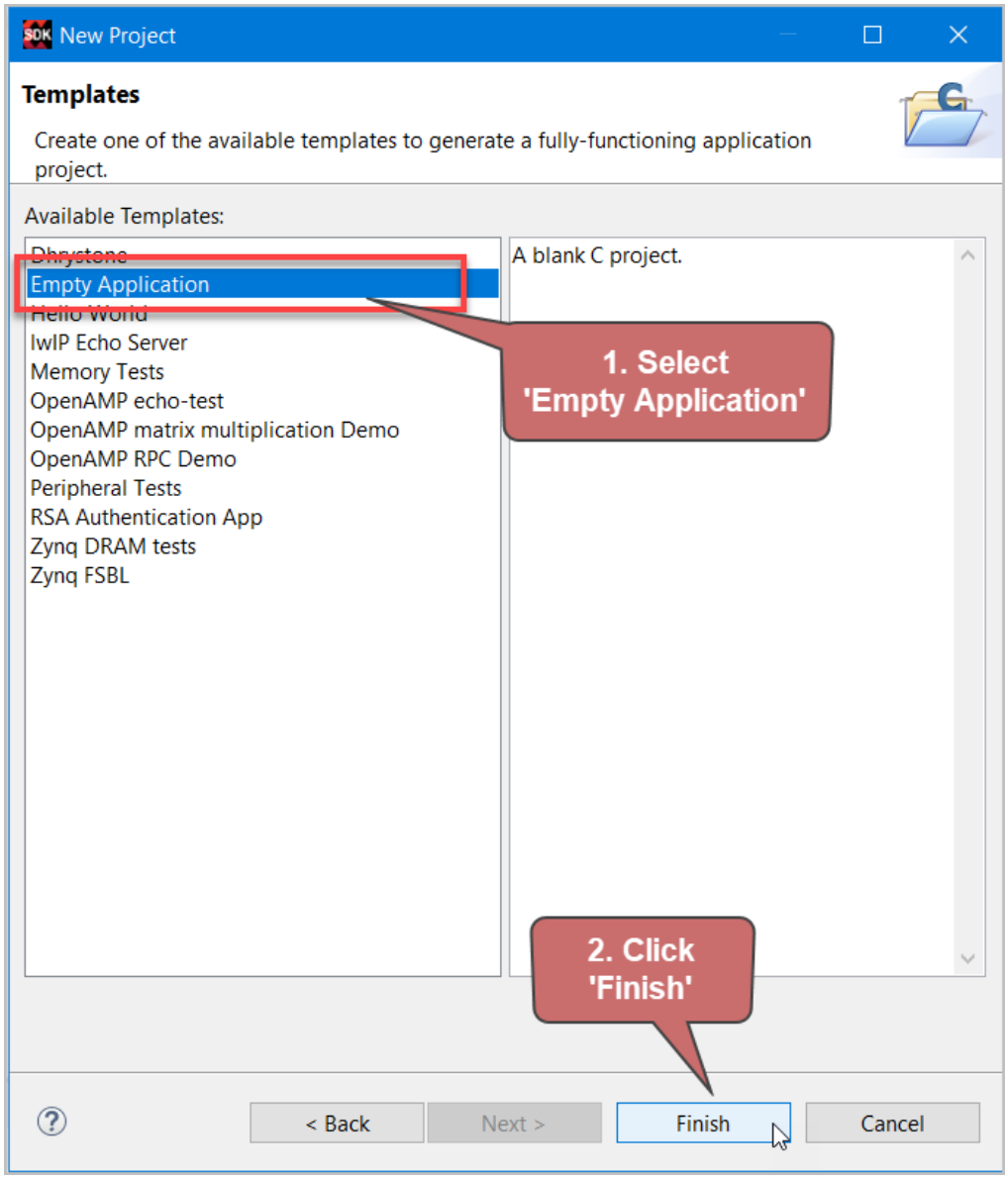


Figure 114. Enter the project details (part 2)

1. Select Empty Application
2. Click Finish

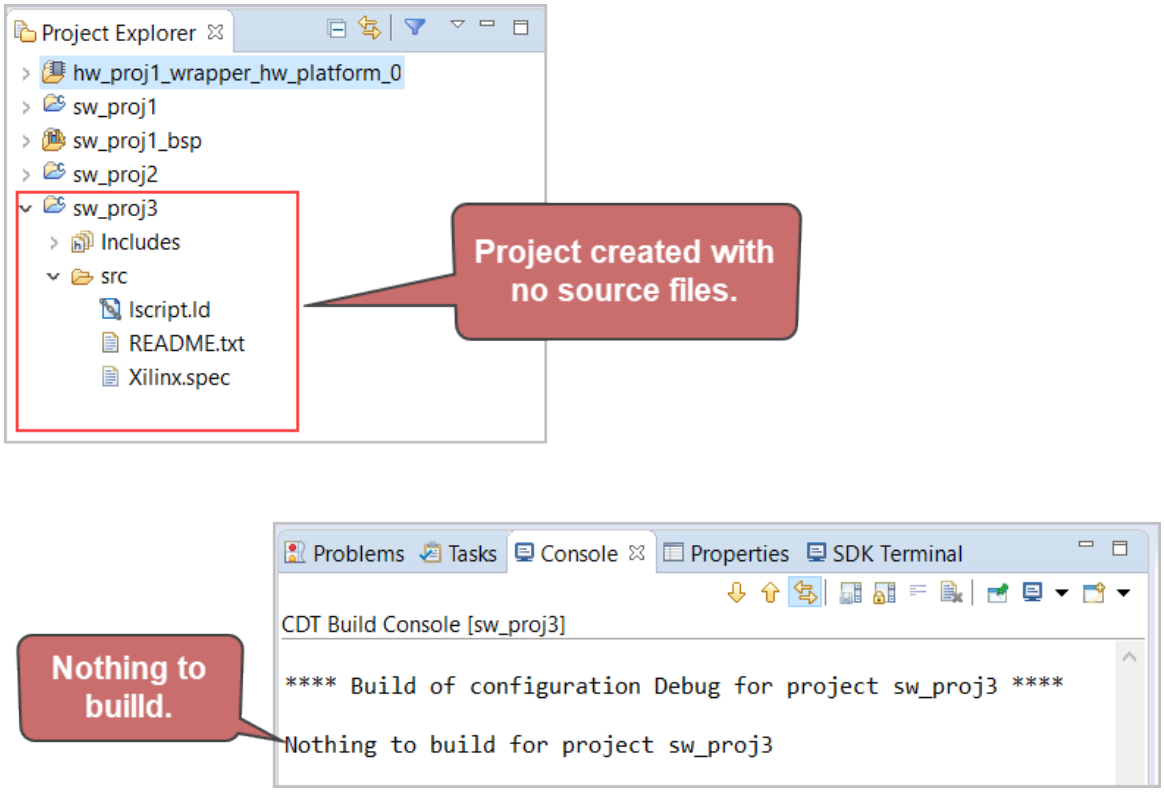


Figure 116. Project created with no source files

A project named "sw_proj3" is created with no project files. The CDT Build Console confirms that there is nothing to build.

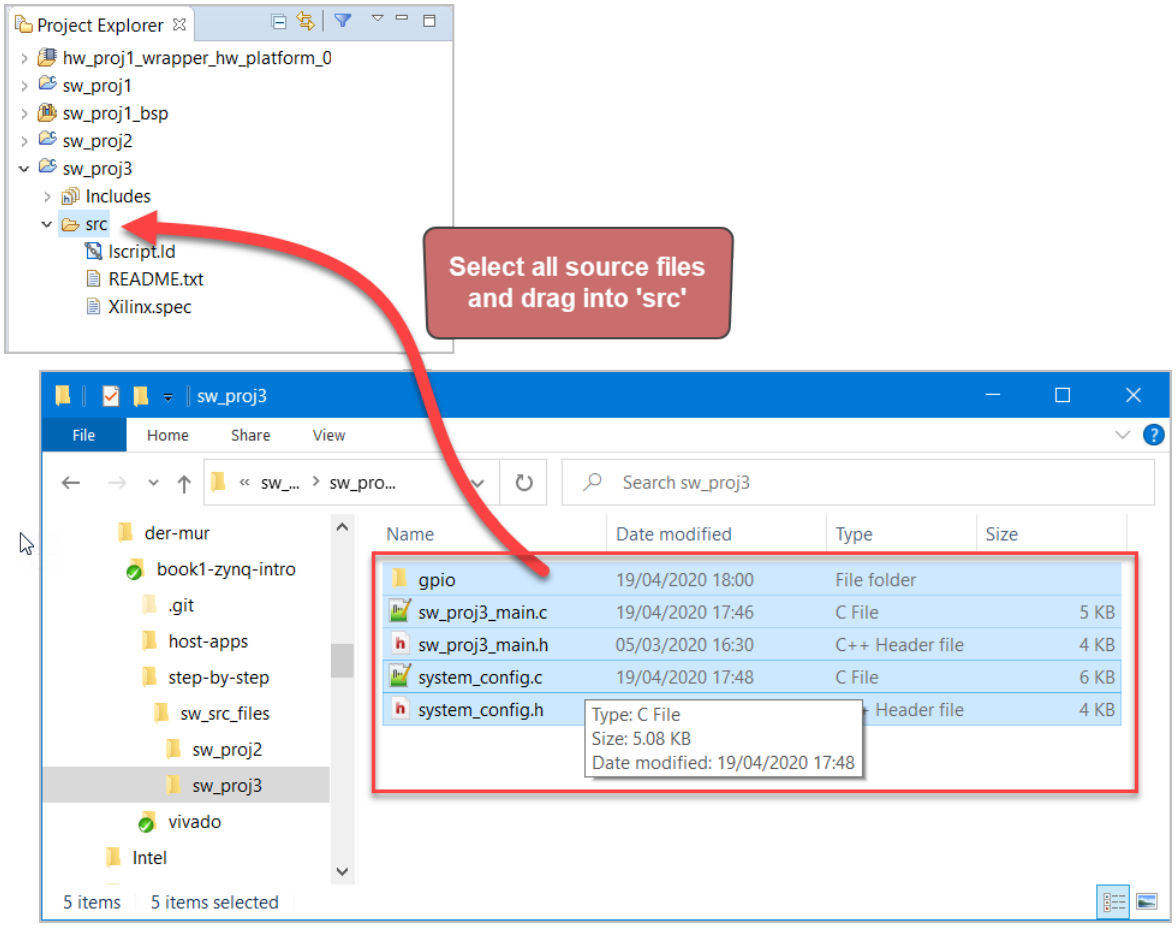


Figure 117. Drag the source files from Windows Explorer into the SDK project

Open the location where the source files for software project 3 are saved on your PC. Drag all the files and directory and from Windows explorer directly into the "**src**" folder in SDK.

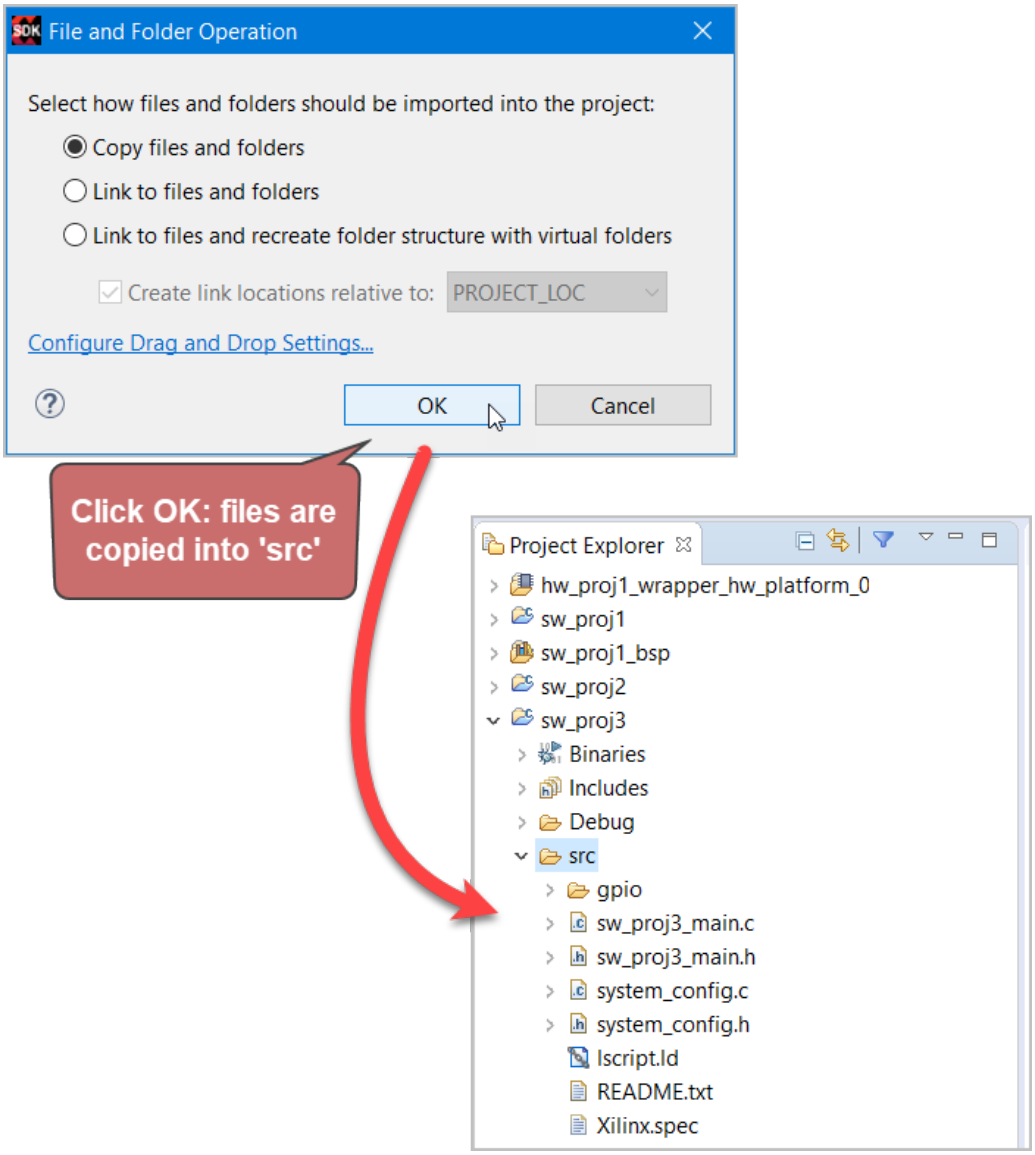


Figure 118. Click OK, and the files will be copied into the project.

In the File Operation dialog box, ensure “Copy files and folders” is checked, and click OK.

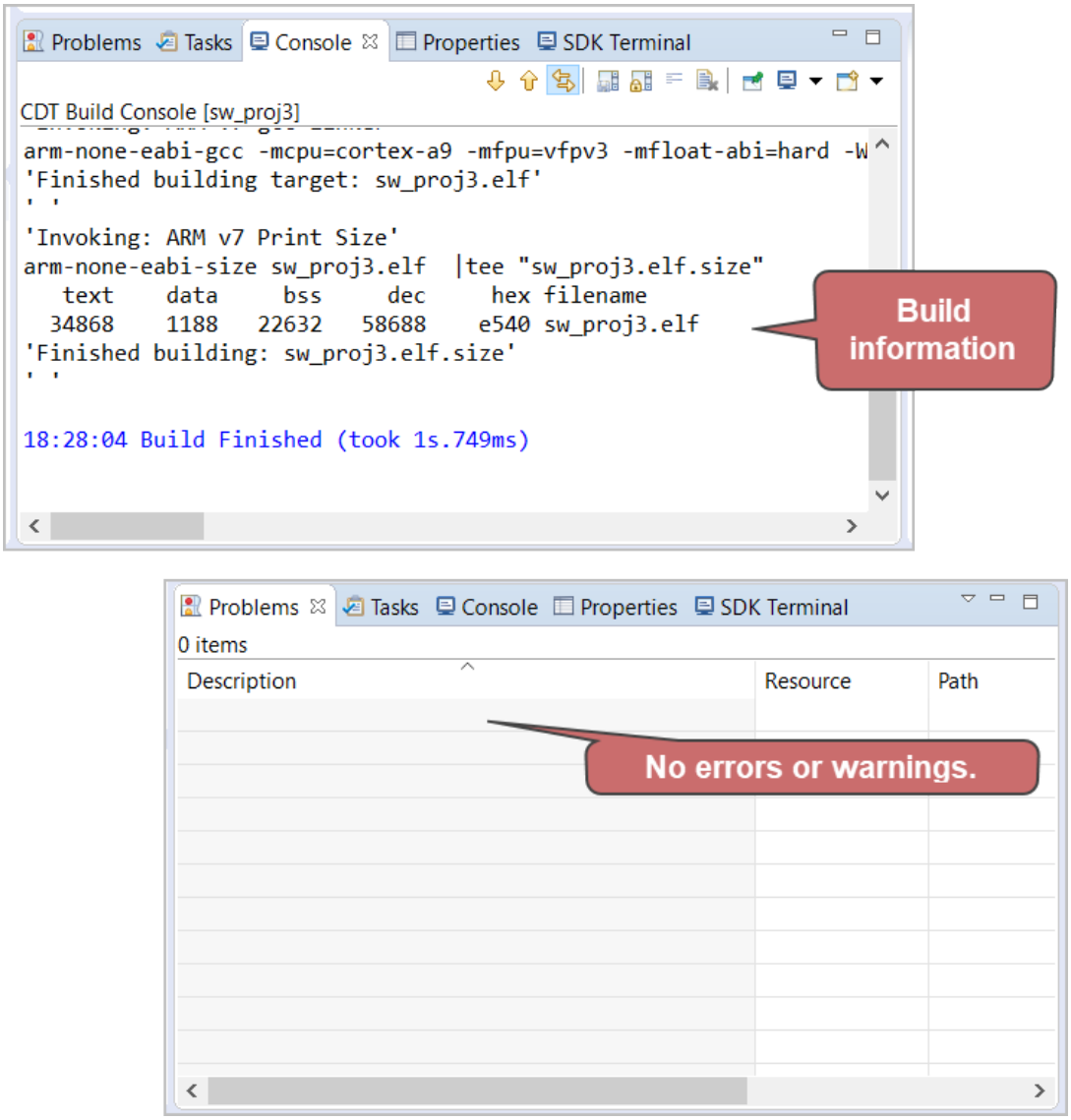


Figure 119. The project should build successfully

If "Build Automatically" is selected, the project should build successfully, with no errors or warnings.

[Exceptions: SDK 2018.3/SDK 2019.1 may indicate a warning as follows, which can be ignored:

#pragma message: For the sleep routines, Global timer is being used

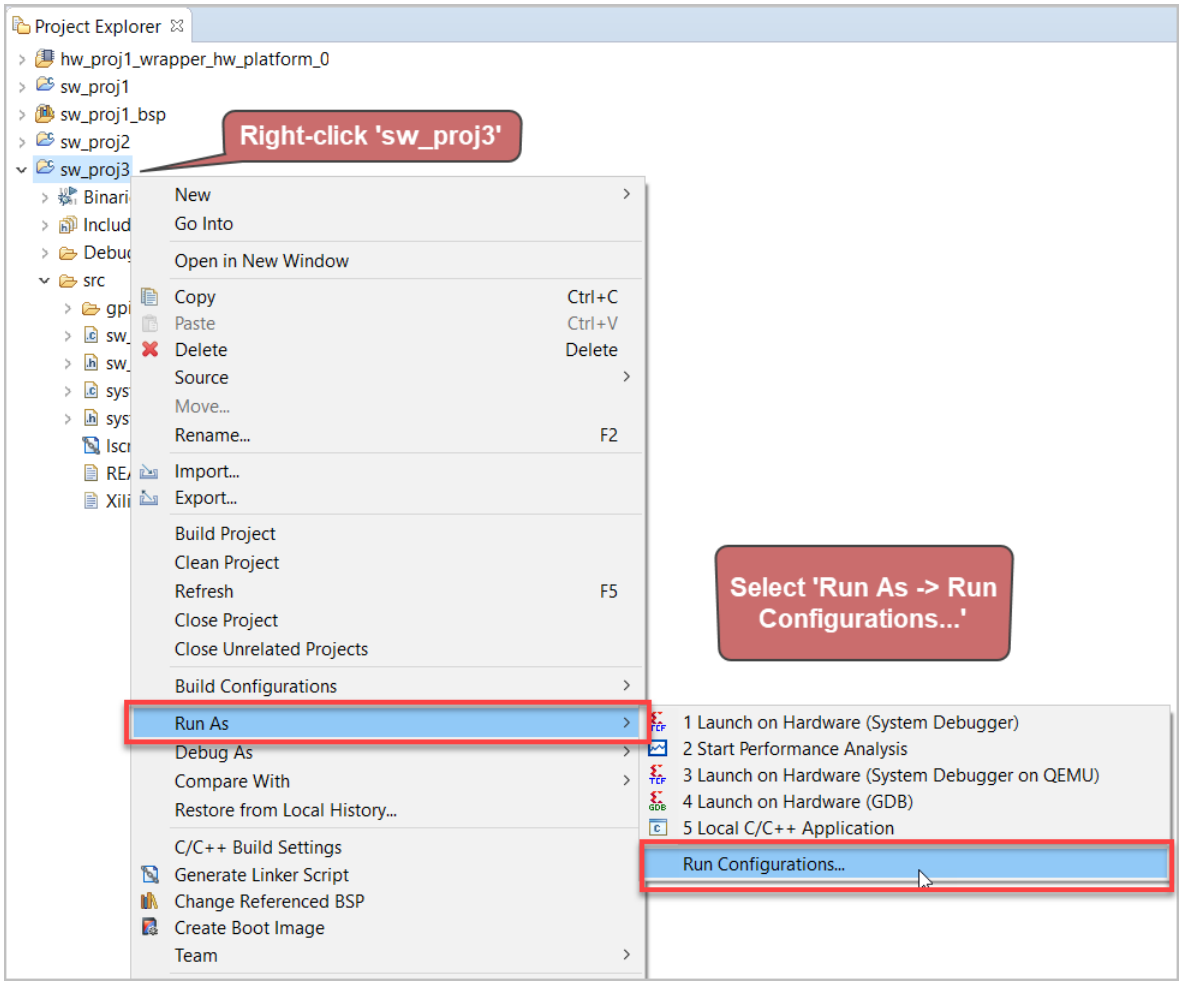


Figure 120. Right-click the software project and create a Run Configuration

To run the program, the first step is to create a Run configuration. Right-click on the project, and select the 'Run Configurations...' option.

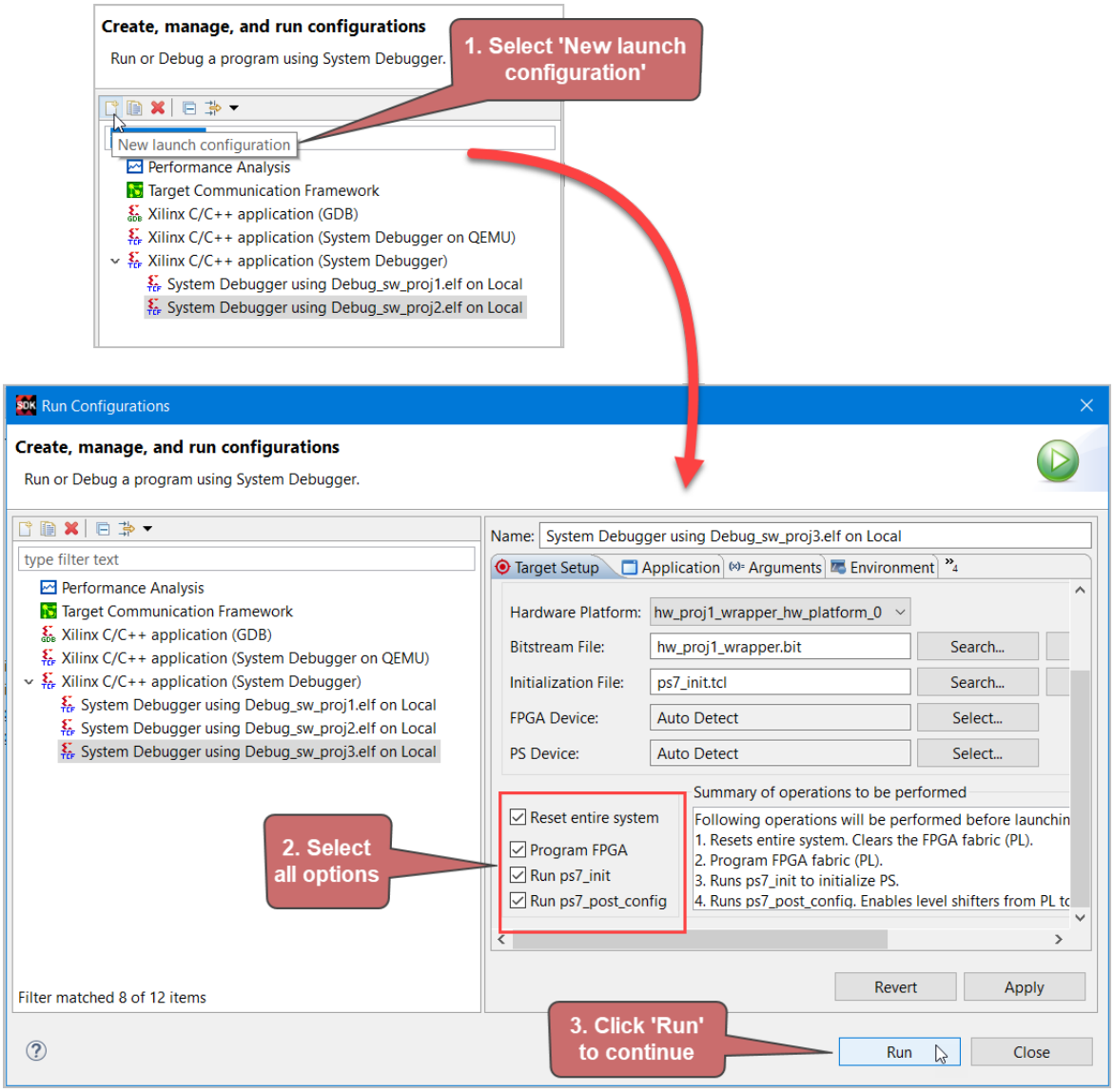


Figure 121. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

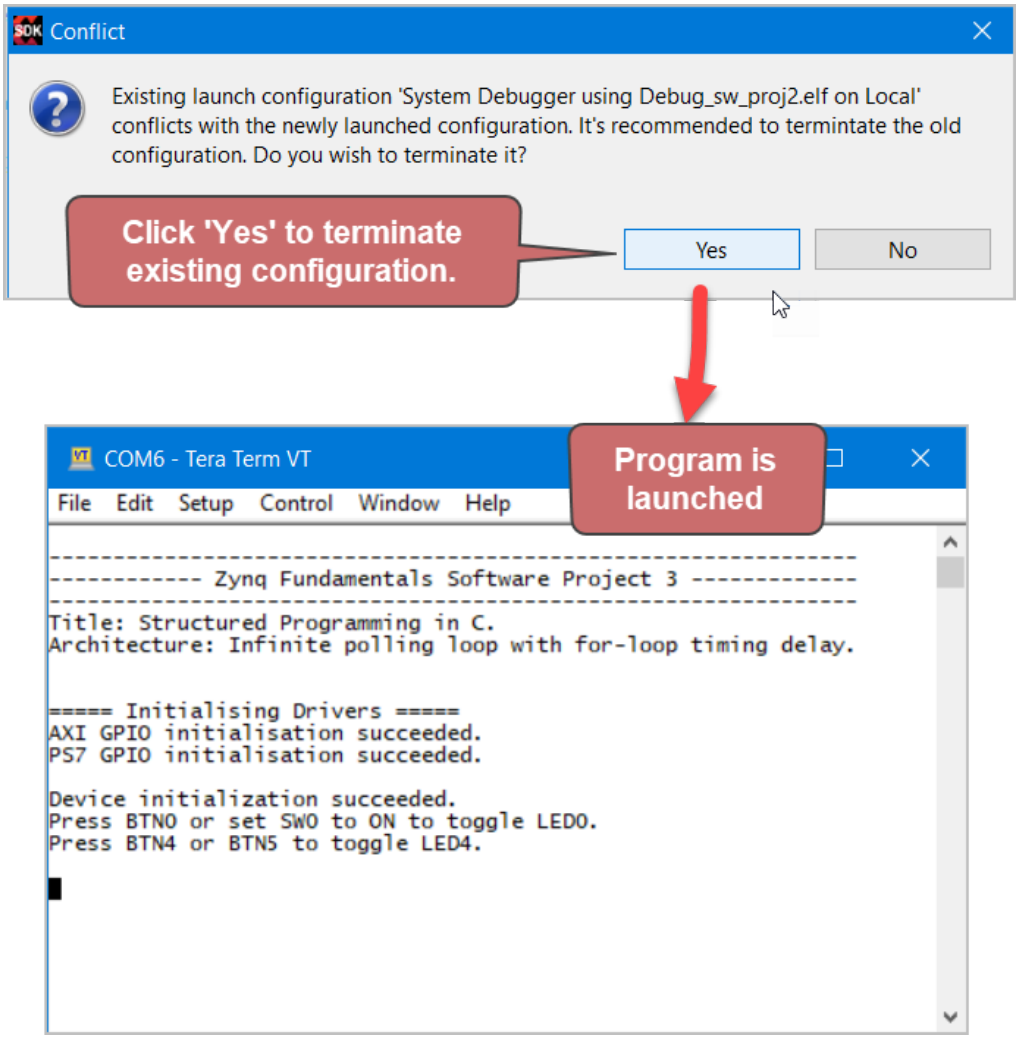


Figure 122. Terminal output for Software Project 3

If prompted, select 'Yes' to terminate any existing configuration. The FPGA will be programmed and the application will be downloaded to the processor. The terminal program should display the results for launching Software Project 3.

3.4 Software Project 4: SCU Timing

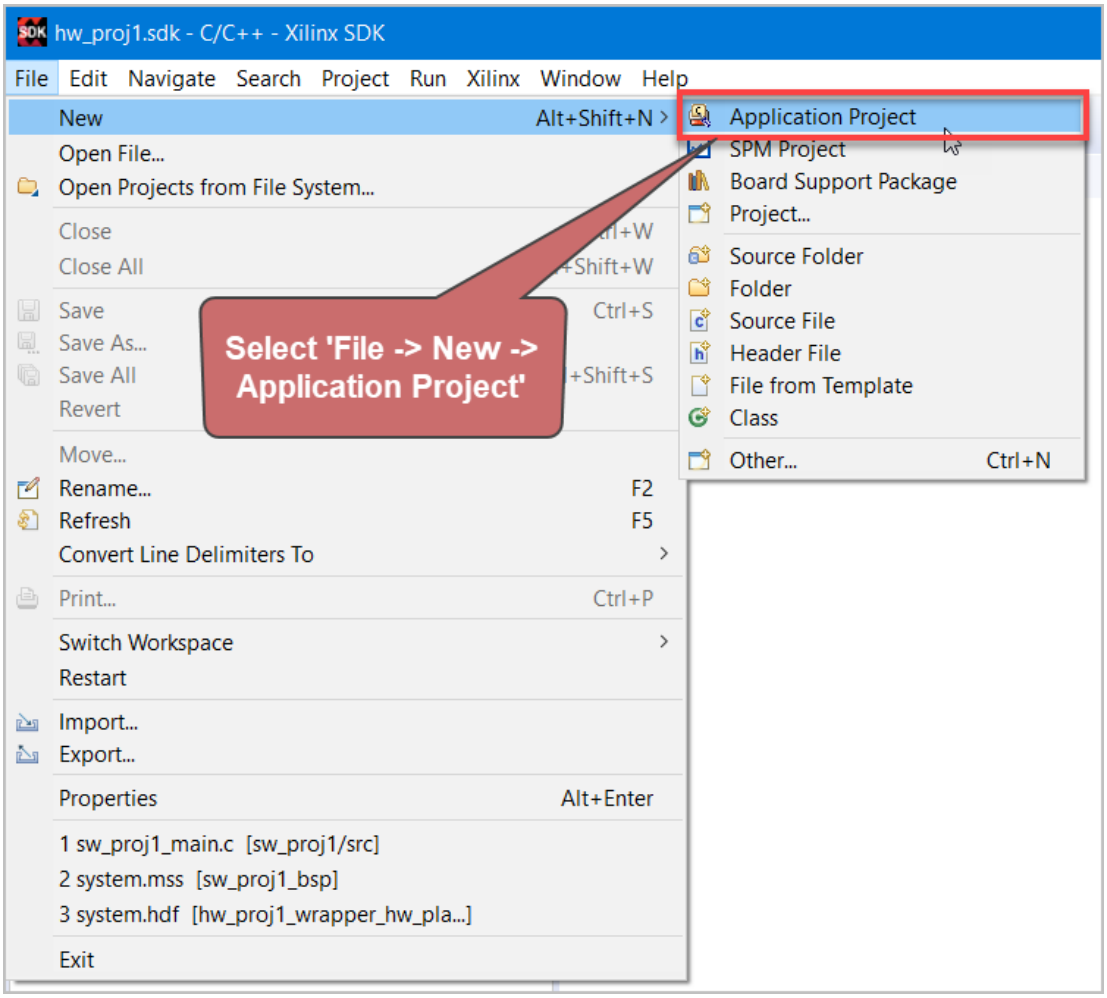


Figure 123. Select File->New-> Application Project

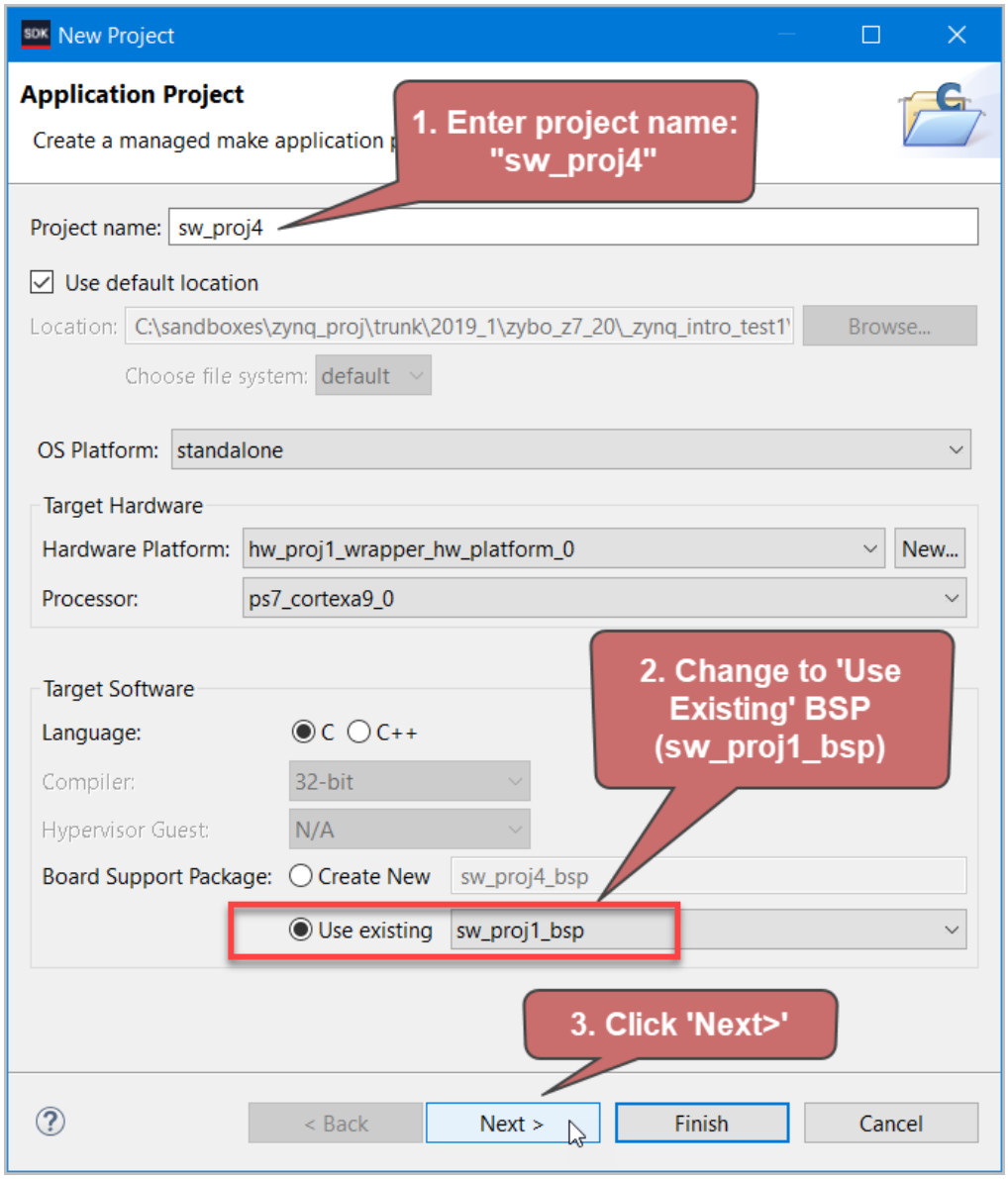


Figure 124. Enter the project details (part 1)

1. Enter the project name: sw_proj4
2. Board Support Package: Use existing
3. Click Next (don't click Finish!)

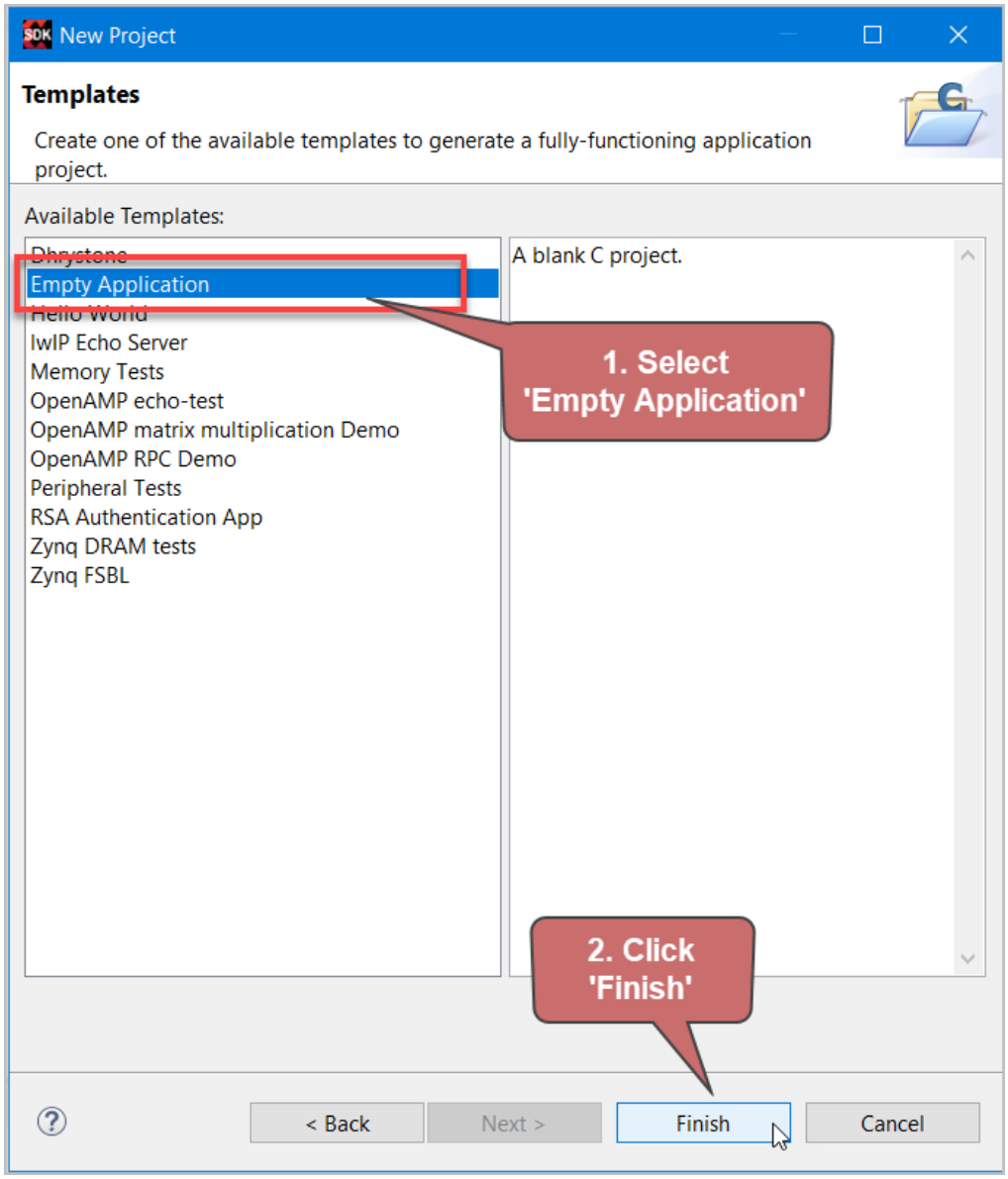


Figure 125. Enter the project details (part 2)

1. Select Empty Application
2. Click Finish

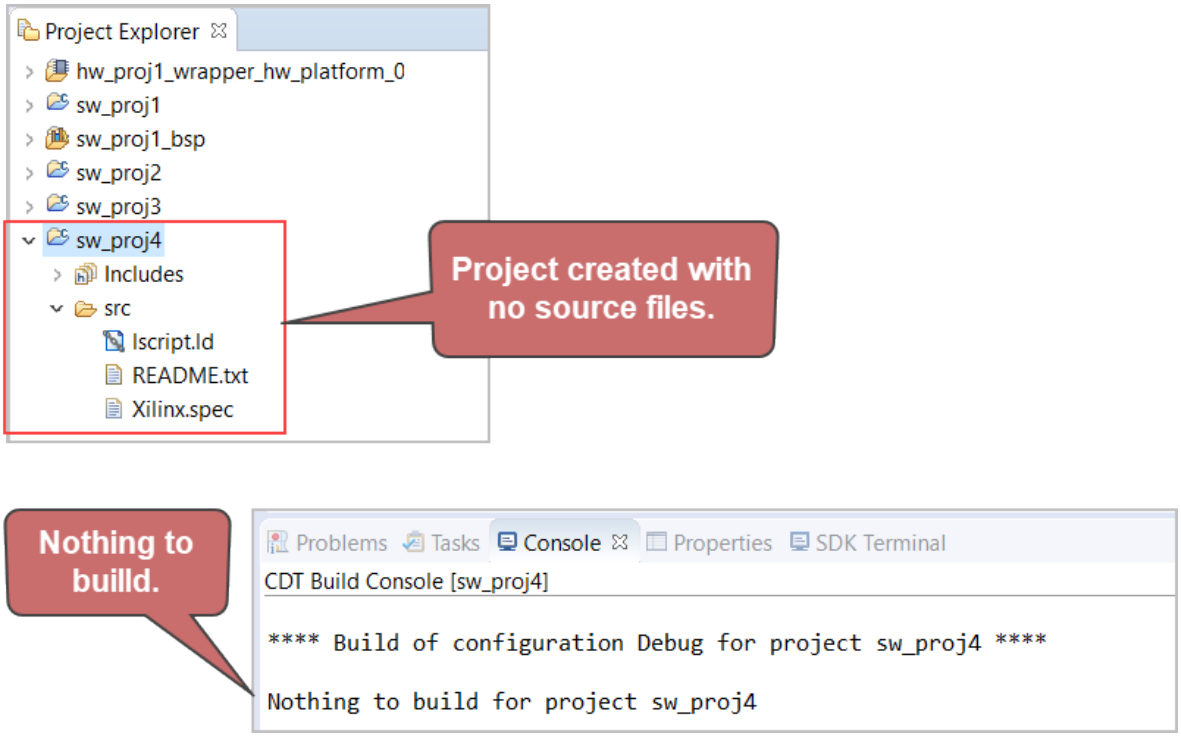


Figure 127. Project created with no source files

A project named “sw_proj4” is created with no project files. The CDT Build Console confirms that there is nothing to build.

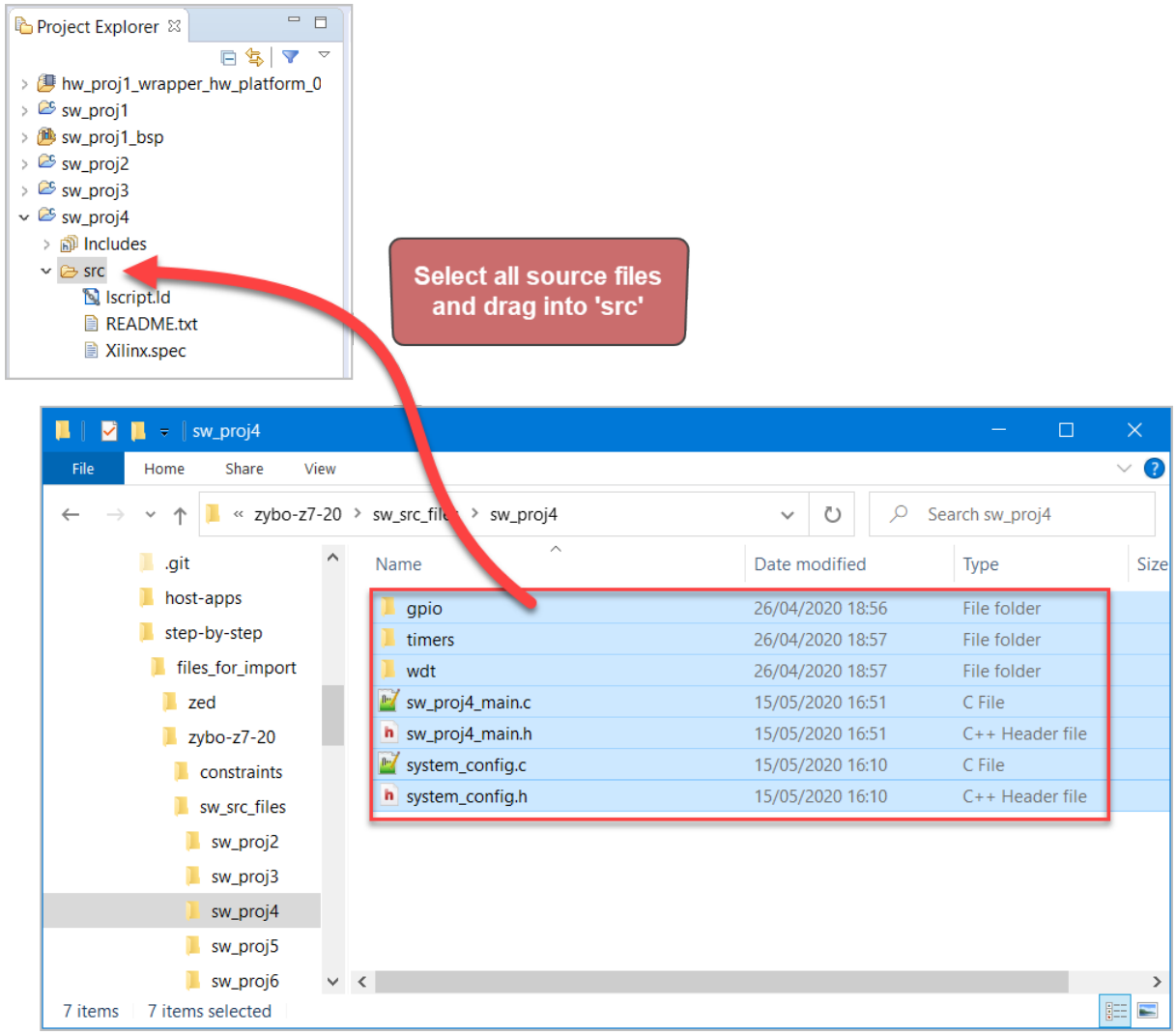


Figure 128. Drag the source files from Windows Explorer into the SDK project

Open the location where the source files for software project 4 are saved on your PC. Drag all the files and directory and from Windows explorer directly into the "**src**" folder in SDK.

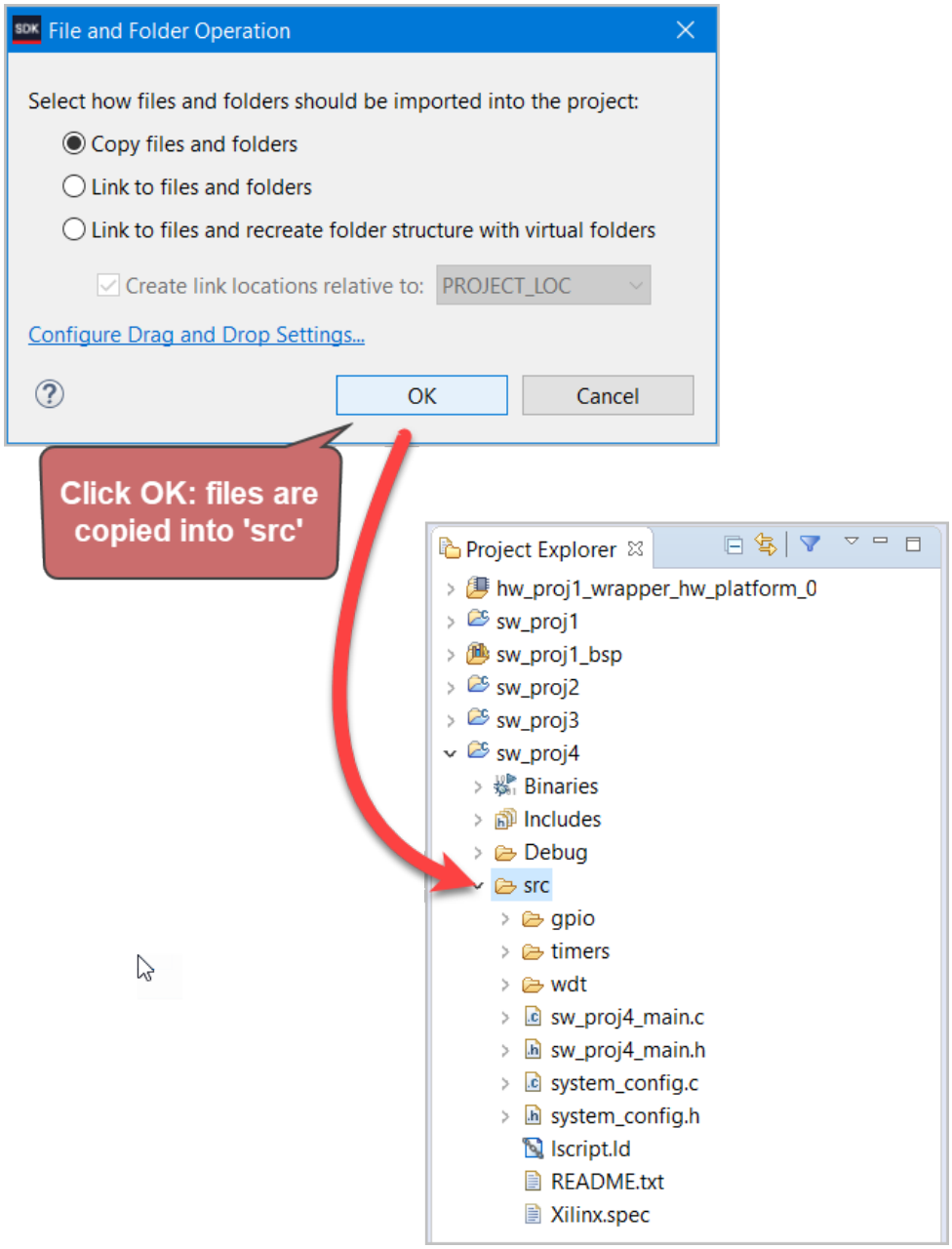


Figure 129. Click OK, and the files will be copied into the project.

In the File Operation dialog box, ensure “Copy files and folders” is checked, and click OK.

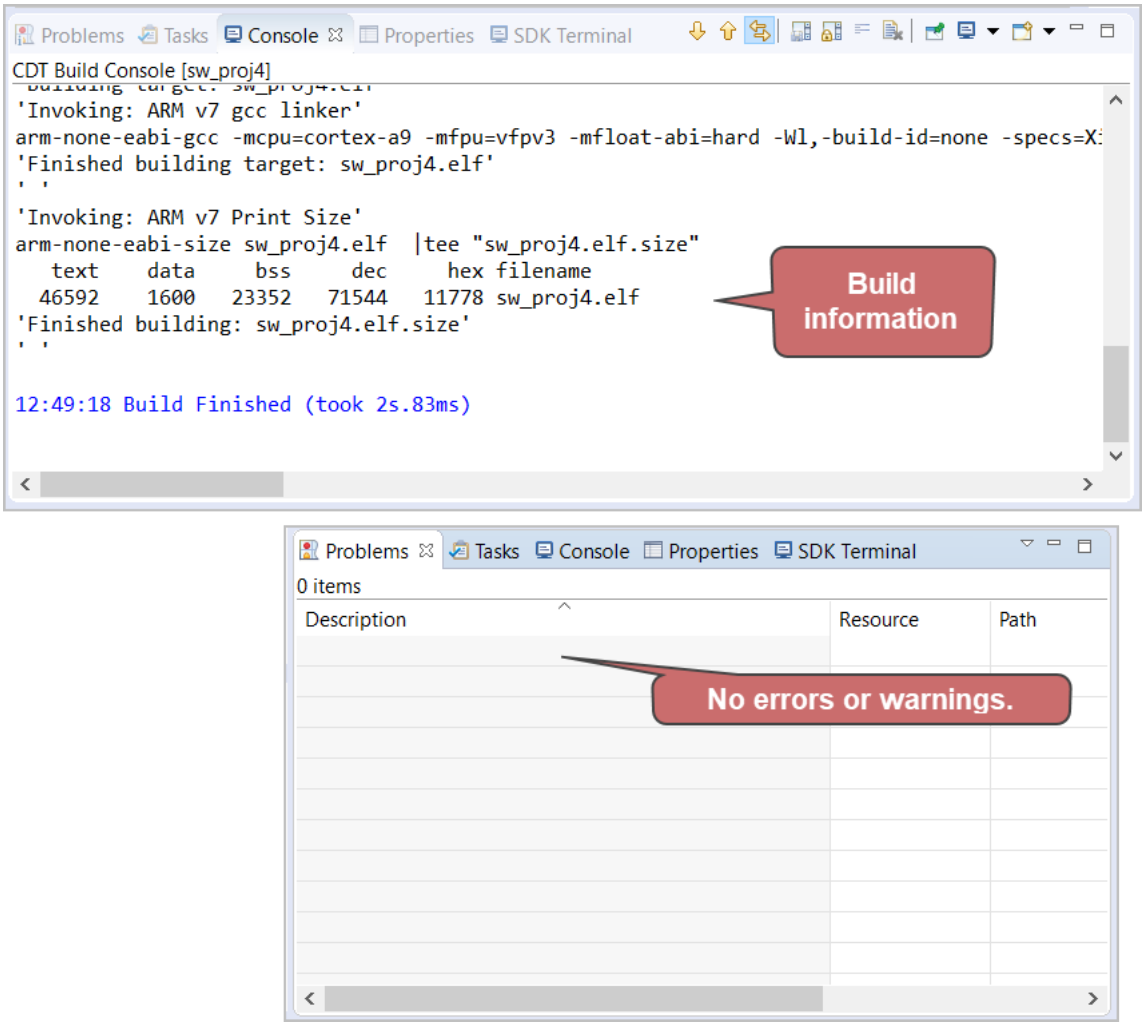


Figure 130. The project should build successfully

If "Build Automatically" is selected, the project should build successfully, with no errors or warnings.

[Exceptions: SDK 2018.3/SDK 2019.1 may indicate a warning as follows, which can be ignored:

#pragma message: For the sleep routines, Global timer is being used

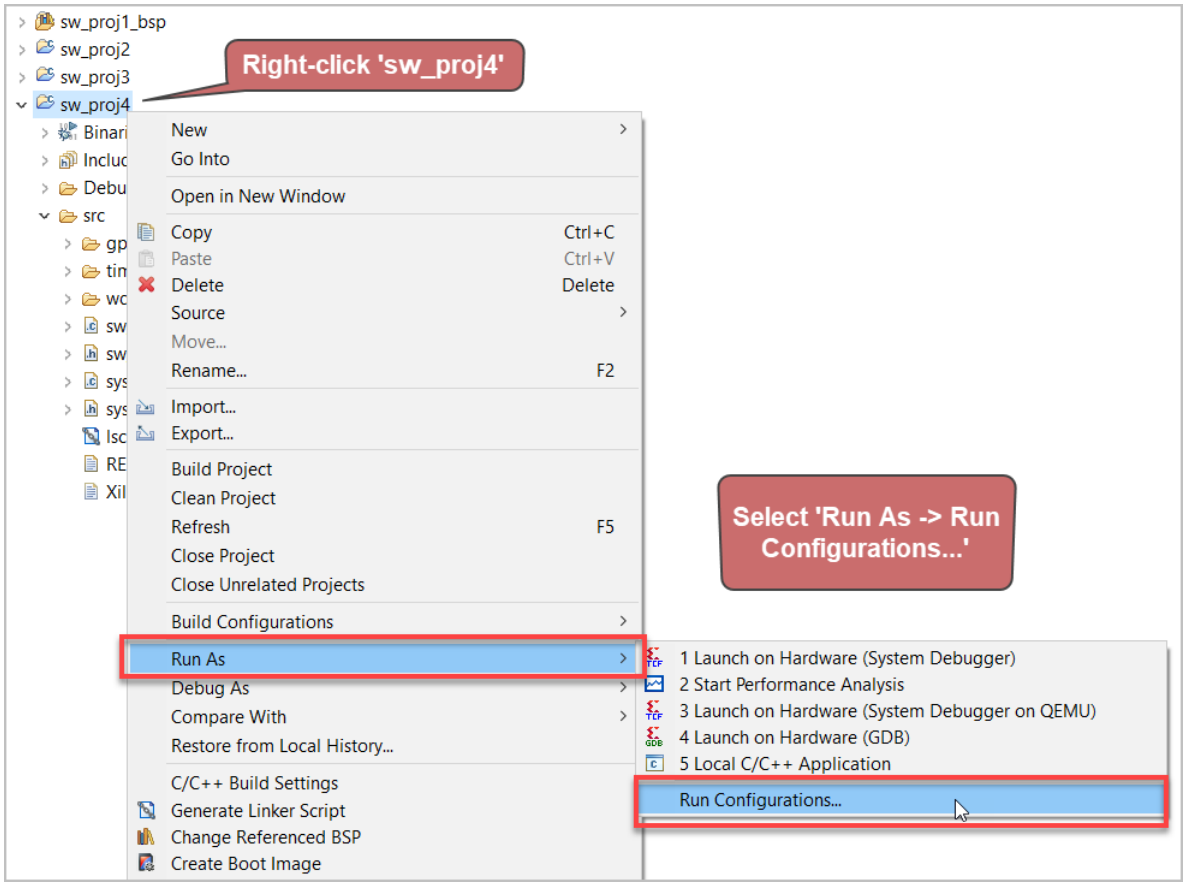


Figure 131. Right-click the software project and create a Run Configuration

To run the program, the first step is to create a Run configuration. Right-click on the project, and select the 'Run Configurations...' option.

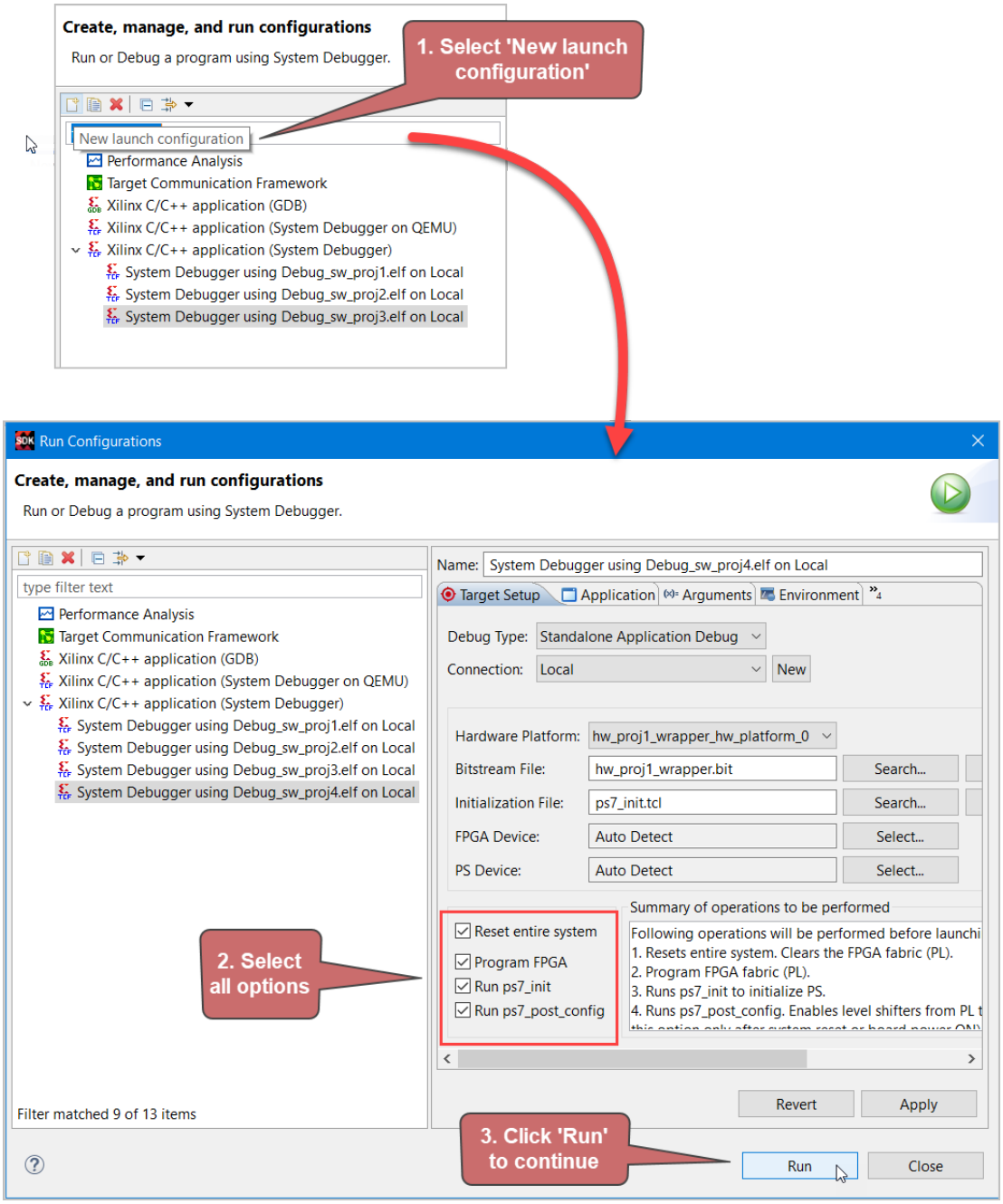


Figure 132. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

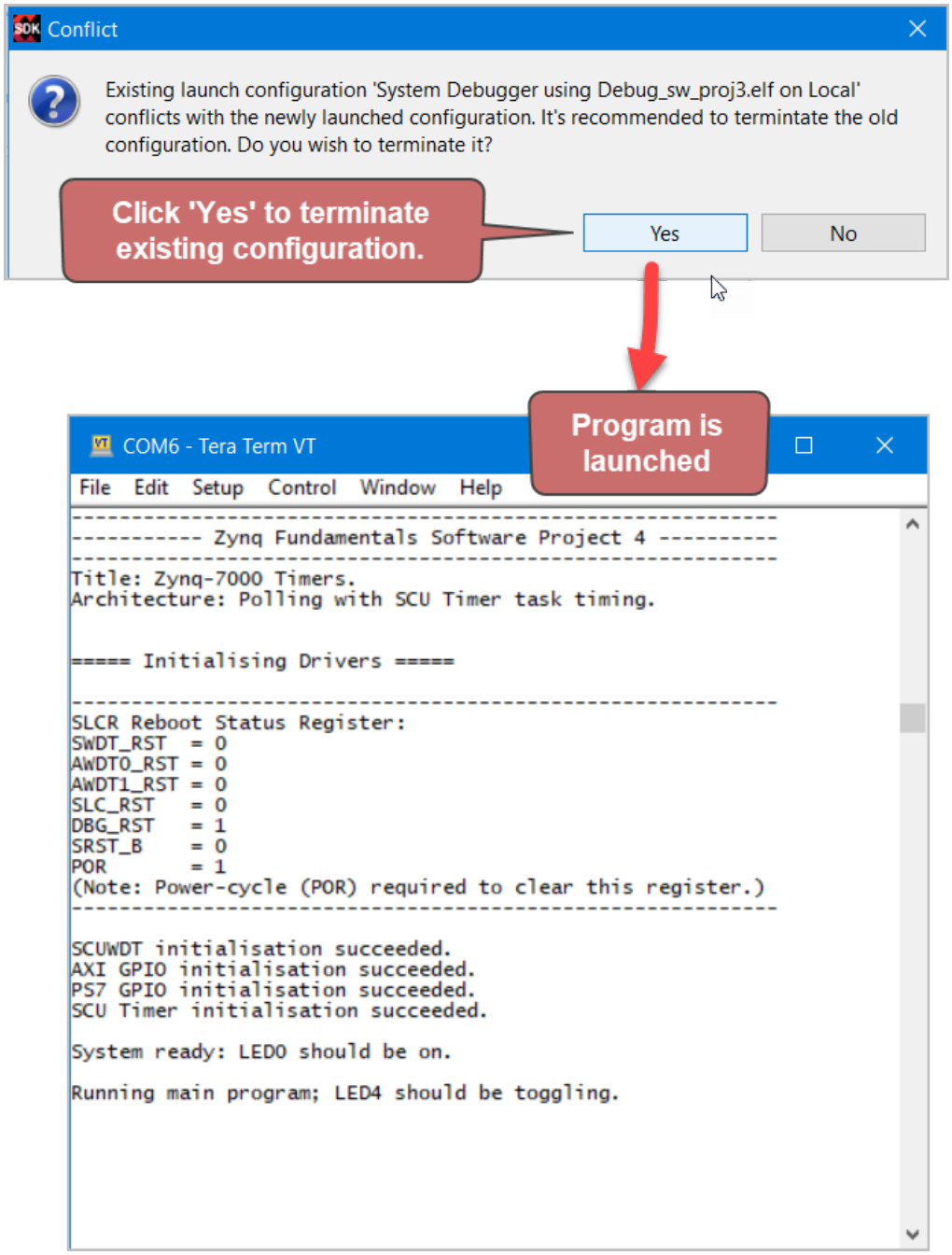


Figure 133. Terminal output for Software Project 4

If prompted, select 'Yes' to terminate any existing configuration. The FPGA will be programmed and the application will be downloaded to the processor. The terminal program should display the results for launching Software Project 4.

3.5 Software Project 5: Zynq Interrupts

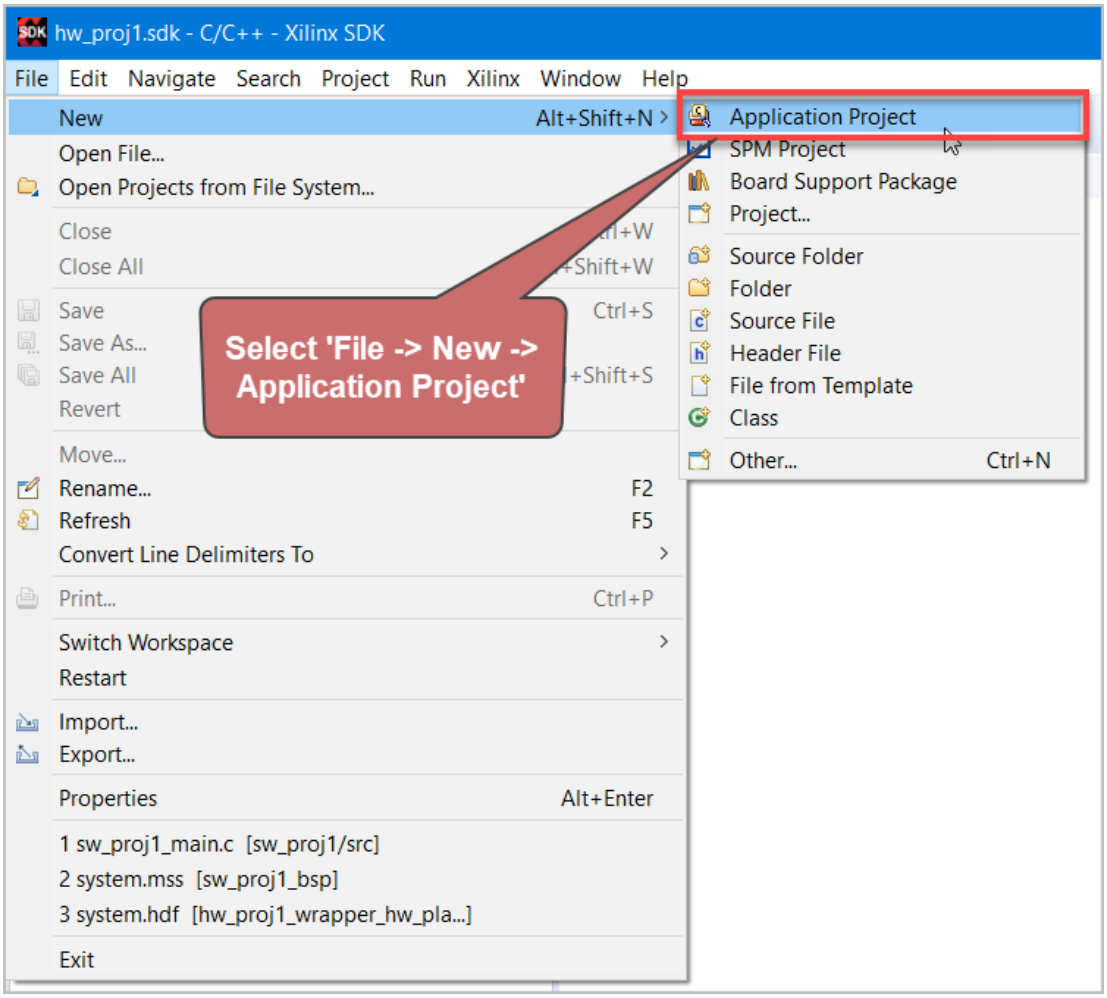


Figure 134. Select File->New-> Application Project

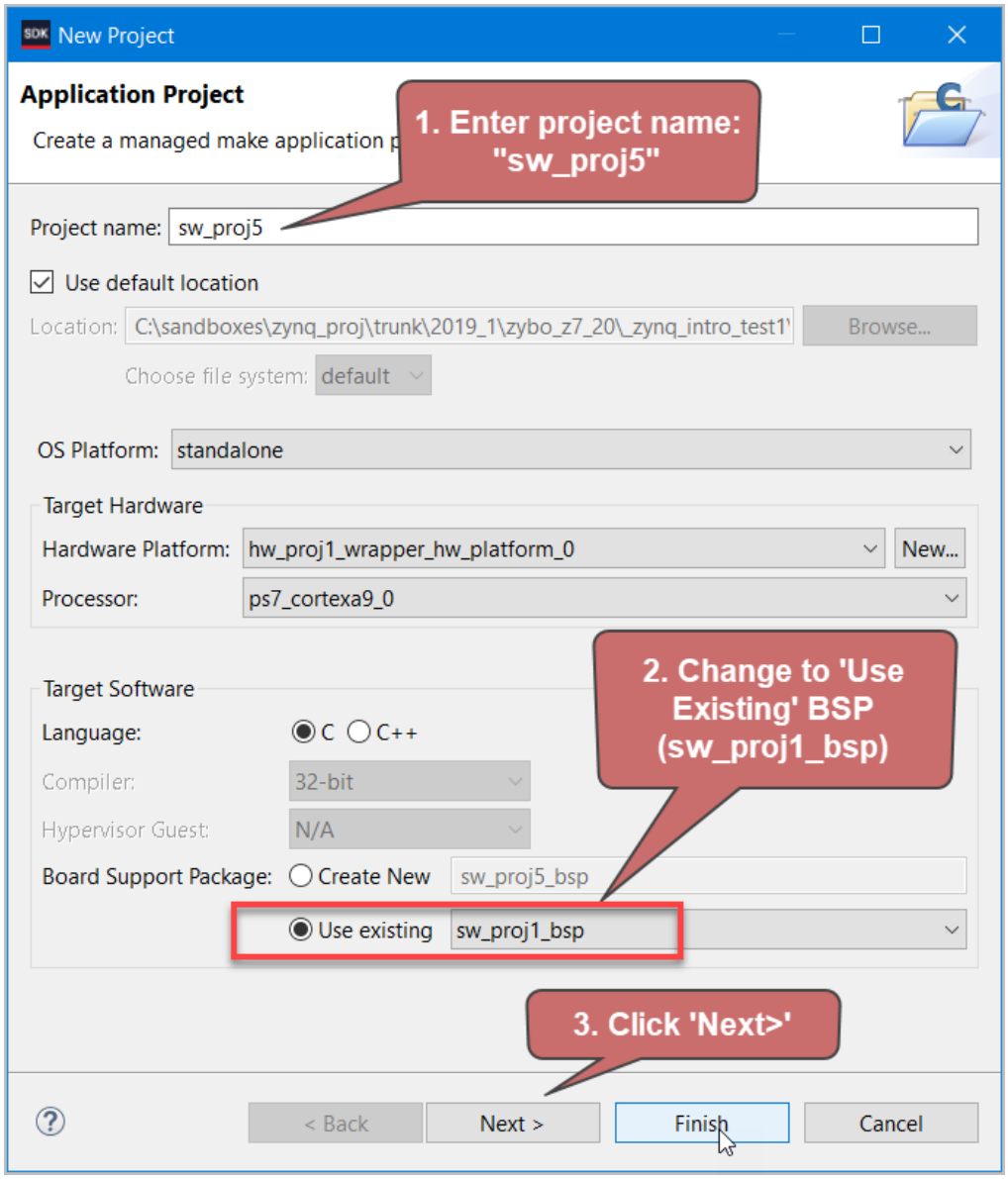


Figure 135. Enter the project details (part 1)

1. Enter the project name: sw_proj5
2. Board Support Package: Use existing
3. Click Next (don't click Finish!)

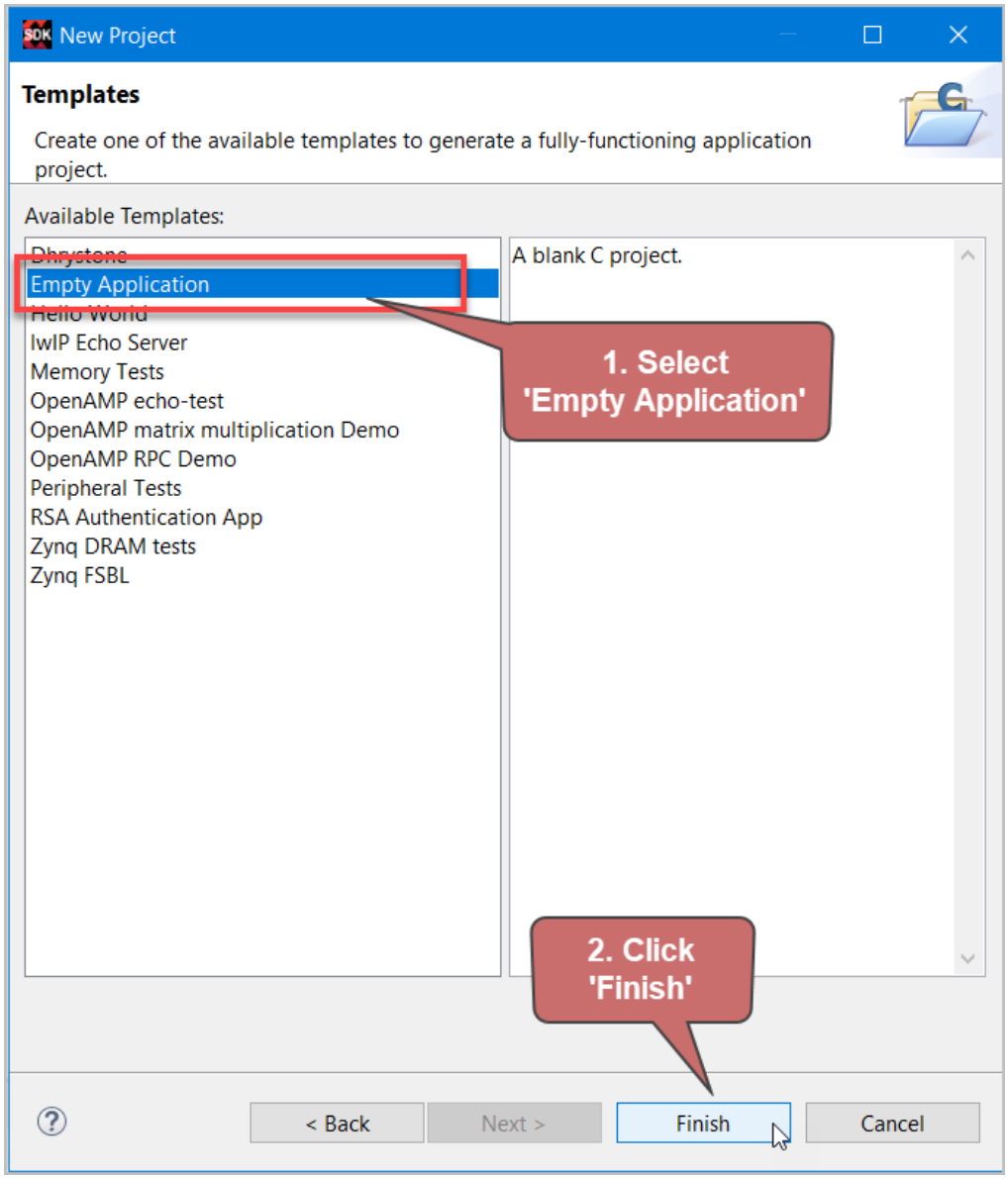


Figure 136. Enter the project details (part 2)

1. Select Empty Application
2. Click Finish

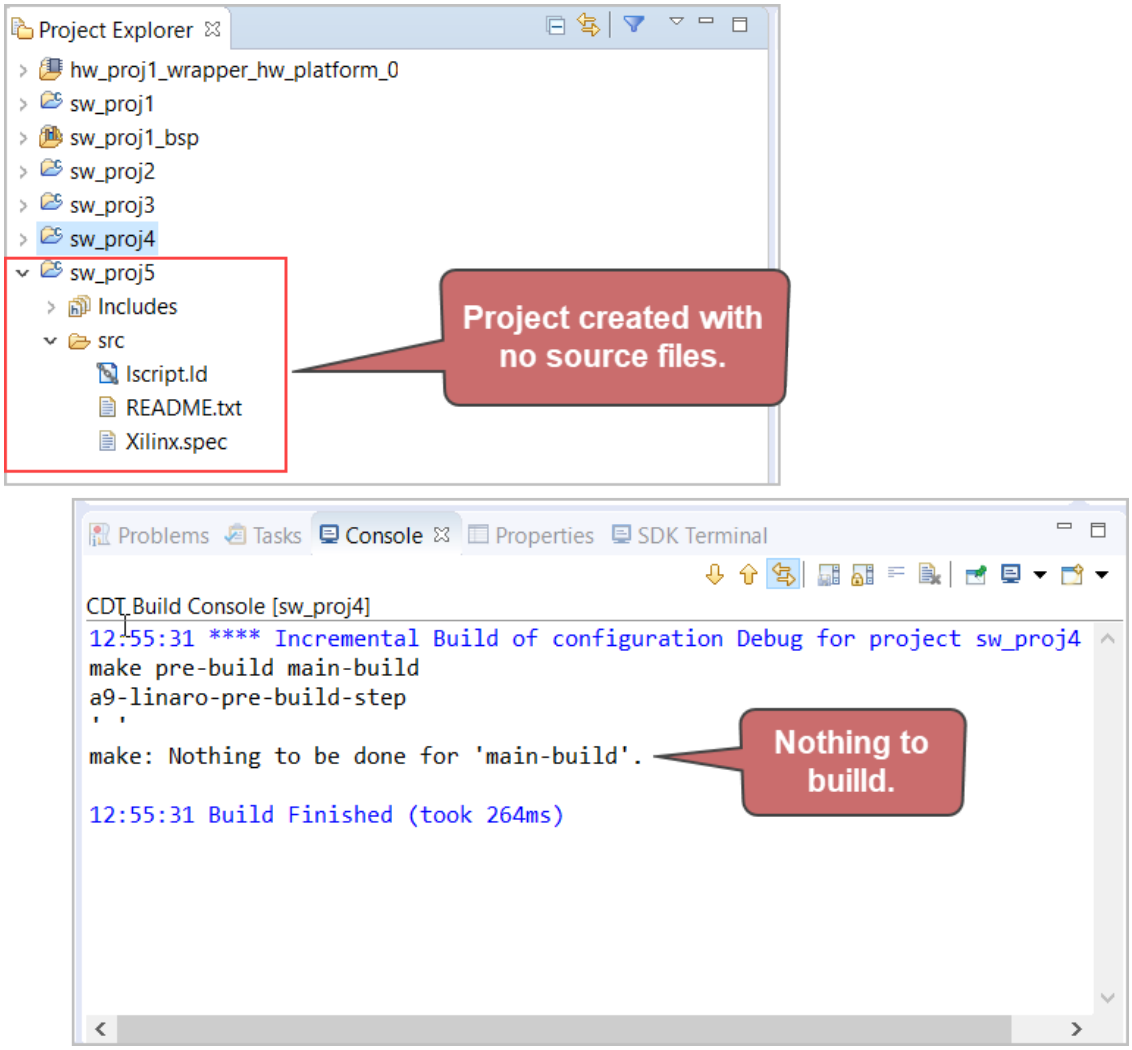


Figure 138. Project created with no source files

A project named "sw_proj5" is created with no project files. The CDT Build Console confirms that there is nothing to build.

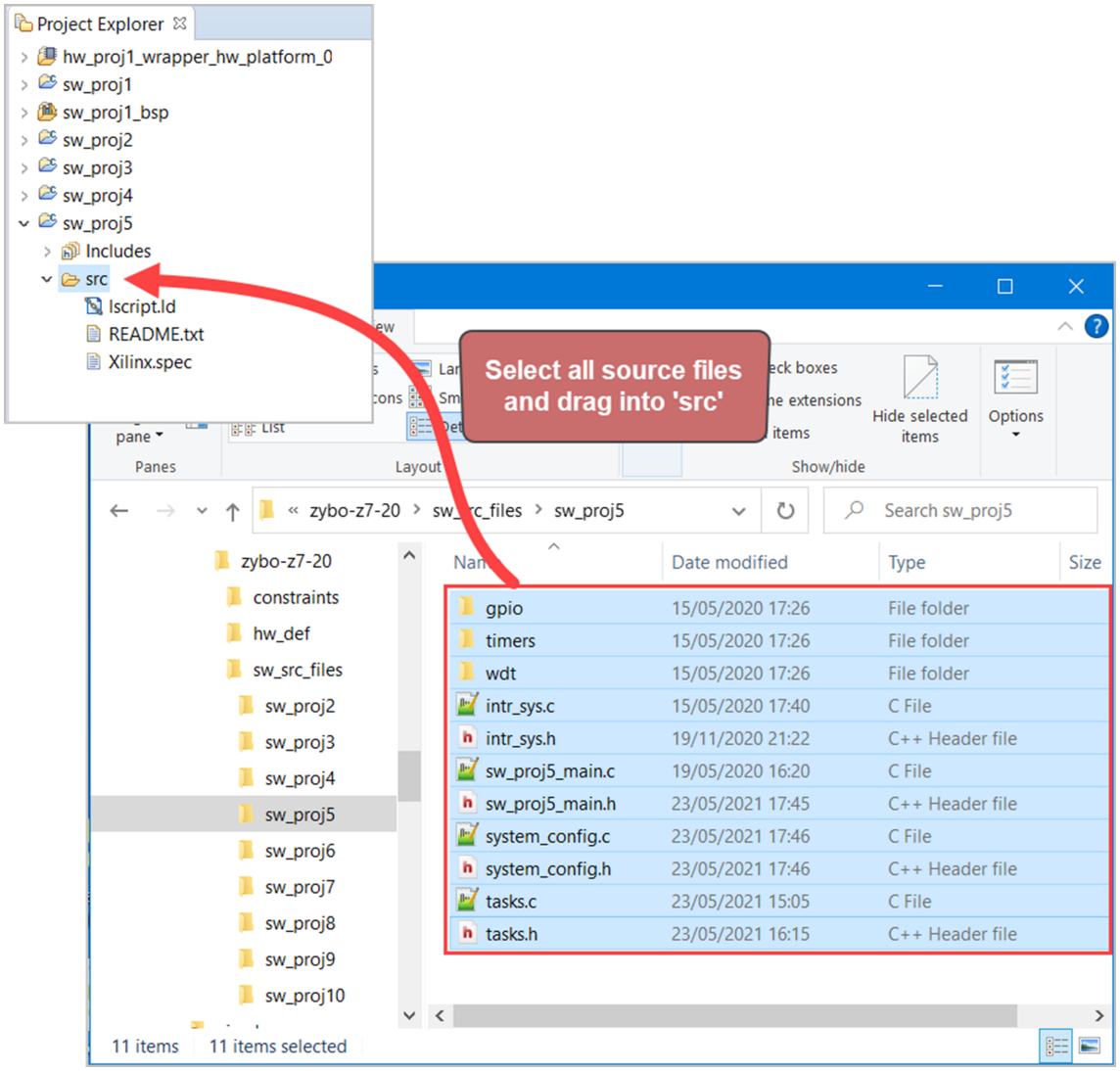


Figure 139. Drag the source files from Windows Explorer into the SDK project

Open the location where the source files for software project 5 are saved on your PC. Drag all the files and directory and from Windows explorer directly into the "**src**" folder in SDK.

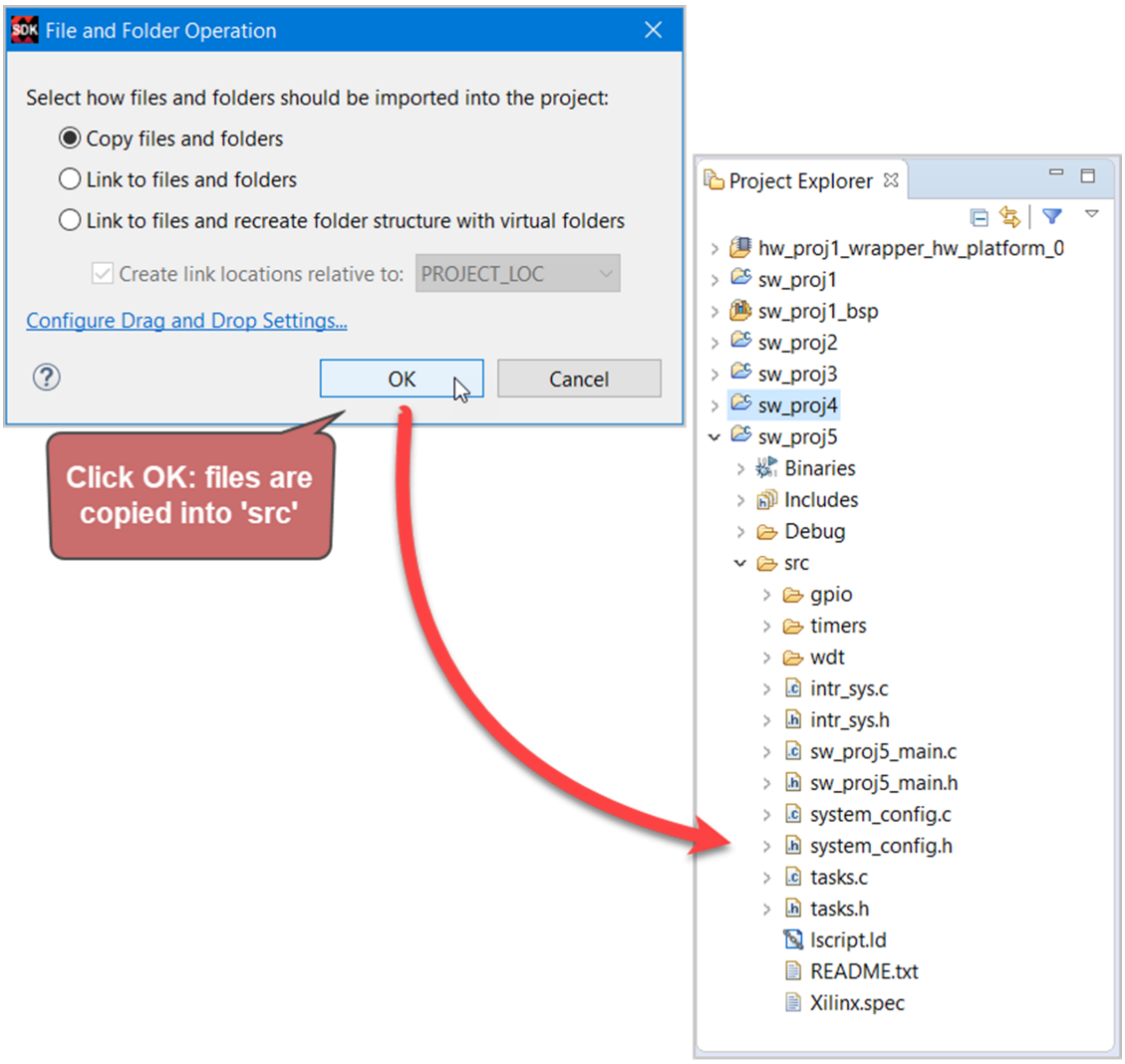


Figure 140. Click OK, and the files will be copied into the project.

In the File Operation dialog box, ensure “Copy files and folders” is checked, and click OK.

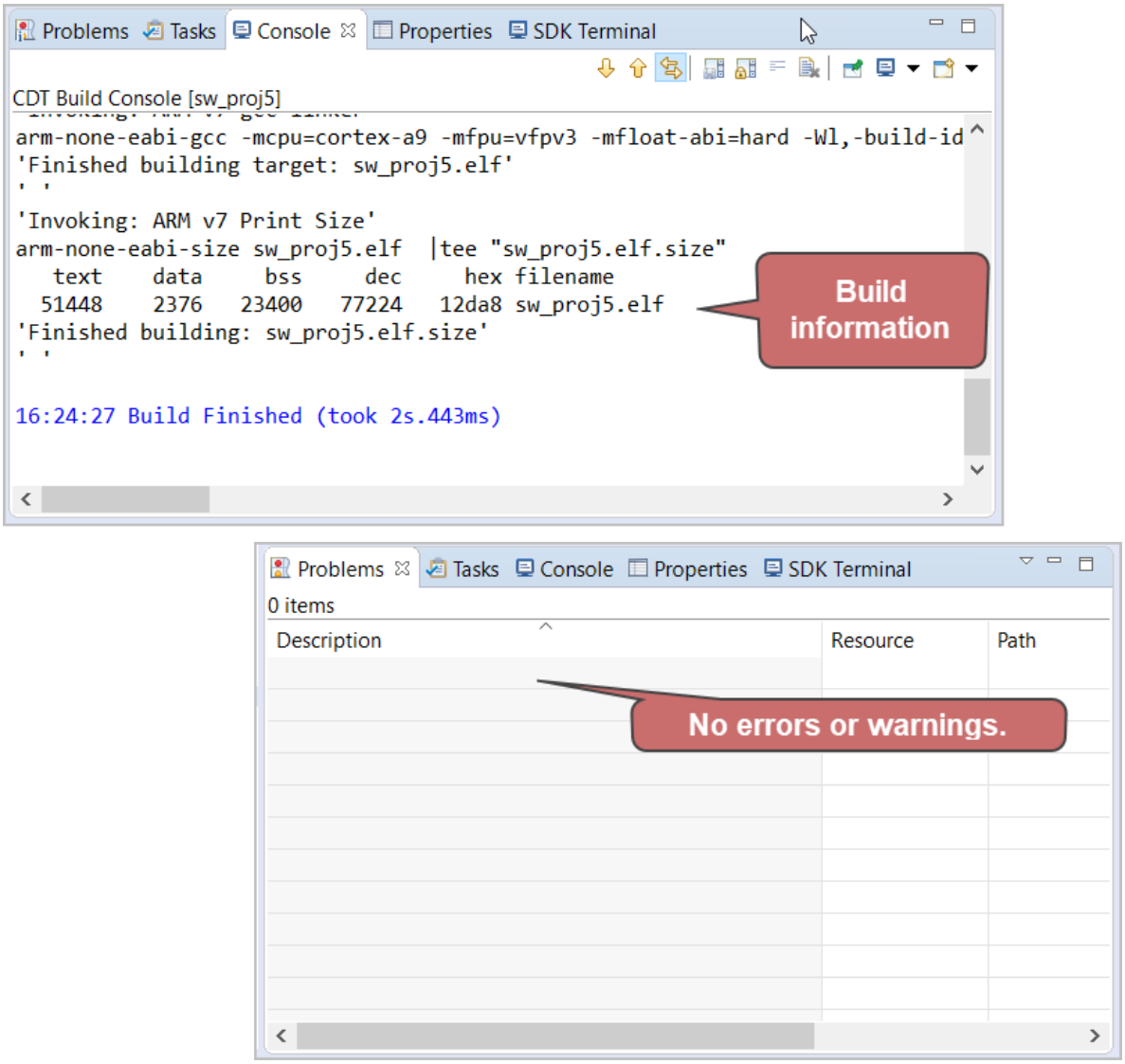


Figure 141. The project should build successfully

If "Build Automatically" is selected, the project should build successfully, with no errors or warnings.

[Exceptions: SDK 2018.3/SDK 2019.1 may indicate a warning as follows, which can be ignored:

#pragma message: For the sleep routines, Global timer is being used

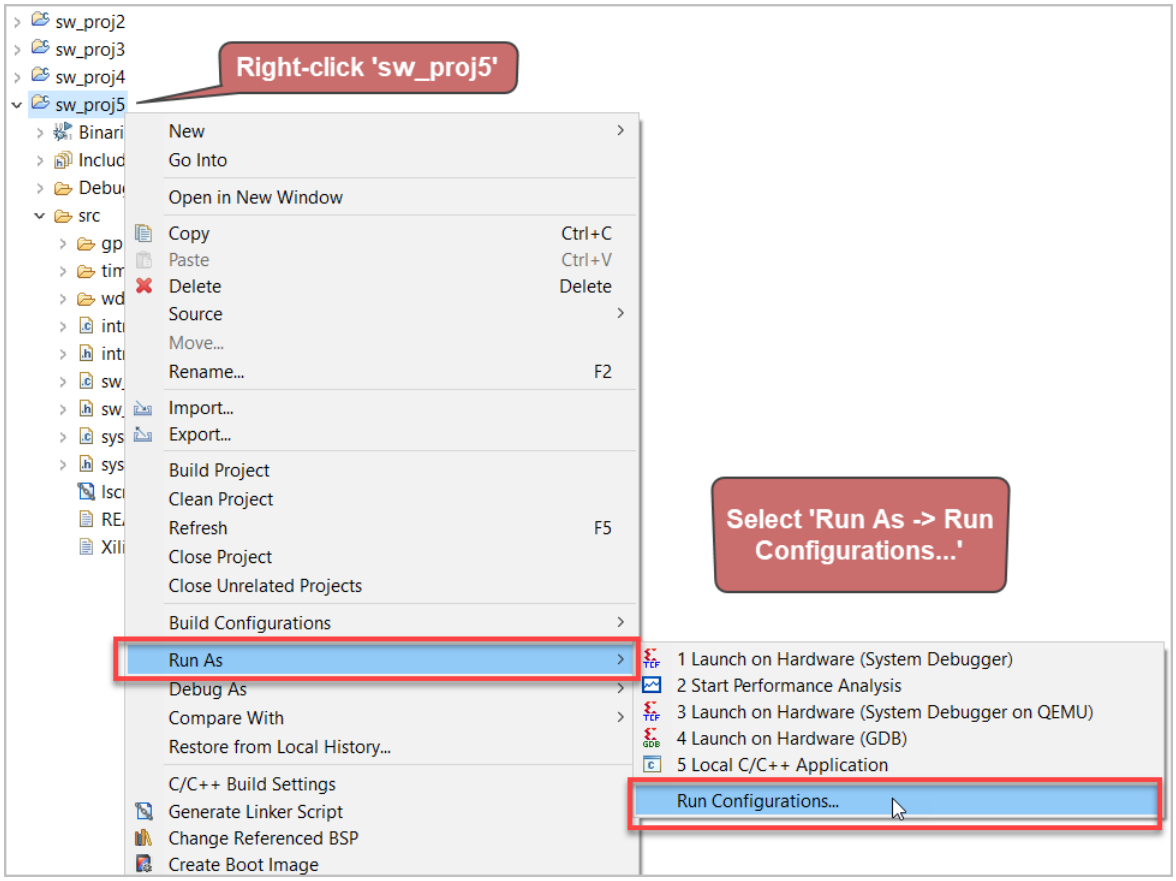


Figure 142. Right-click the software project and create a Run Configuration

To run the program, the first step is to create a Run configuration. Right-click on the project, and select the 'Run Configurations...' option.

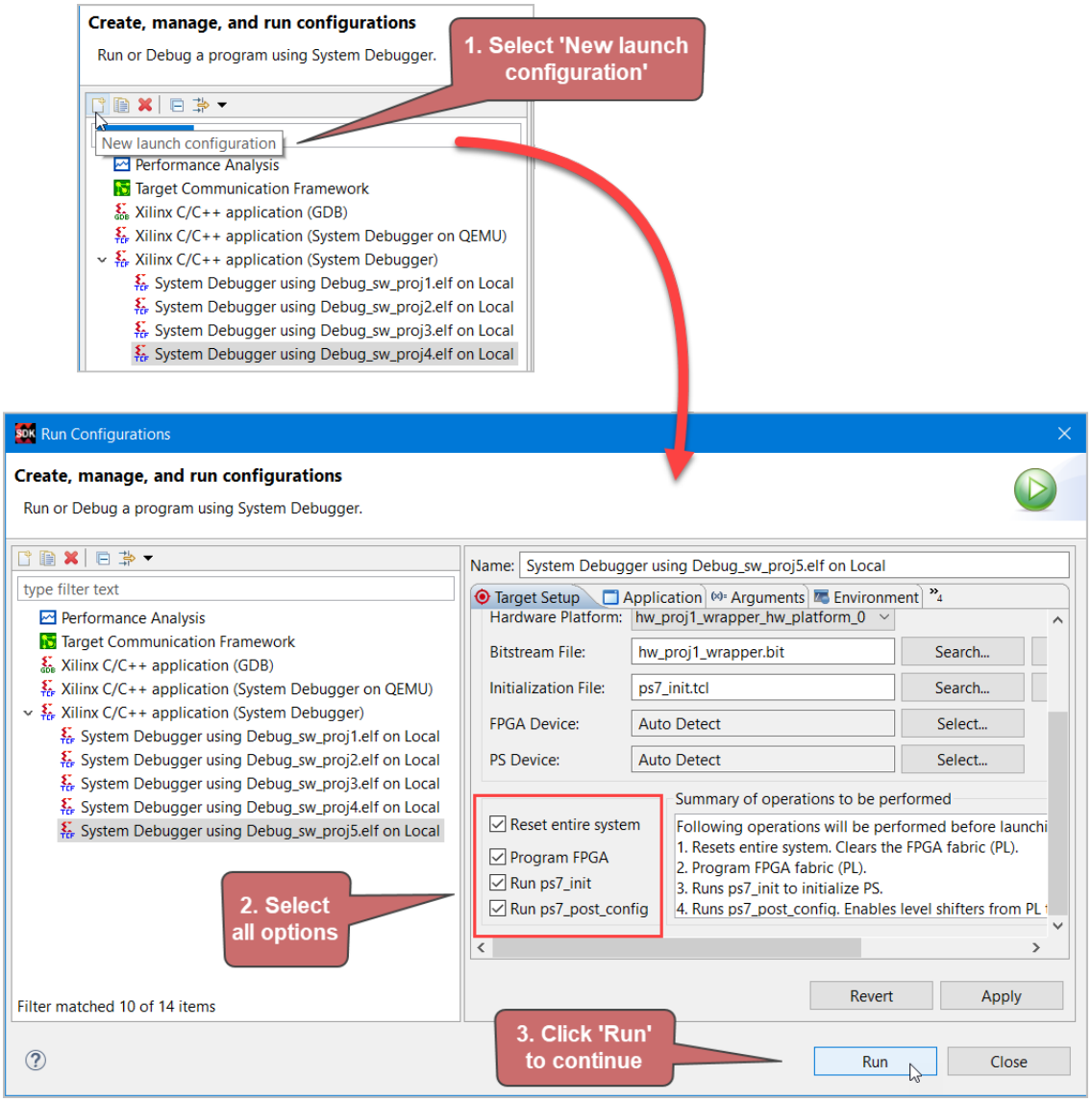


Figure 143. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

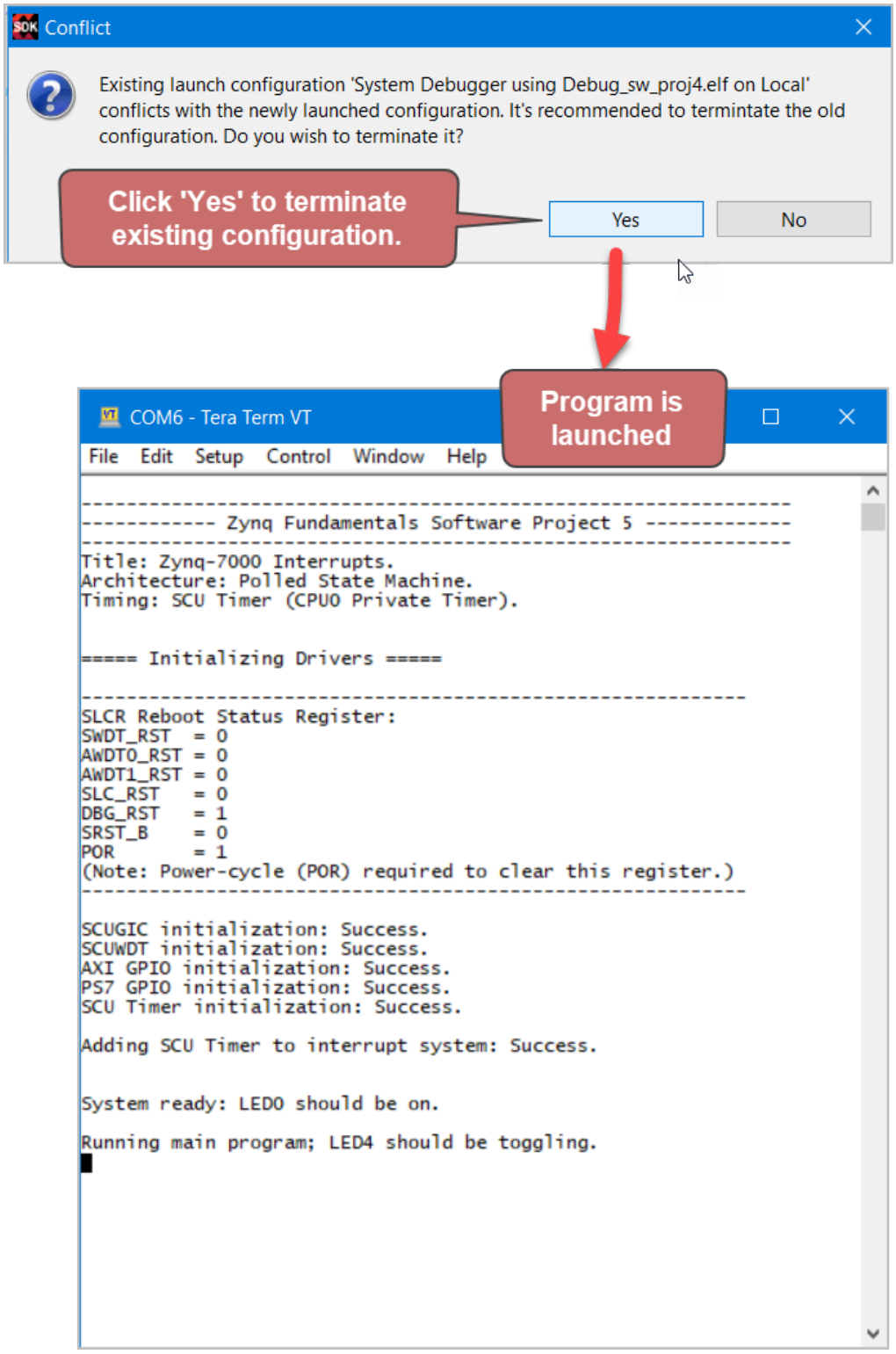


Figure 144. Terminal output for Software Project 5

If prompted, select 'Yes' to terminate any existing configuration. The FPGA will be programmed and the application will be downloaded to the processor. The terminal program should display the results for launching Software Project 5.

3.6 Software Project 6: TTC Timing

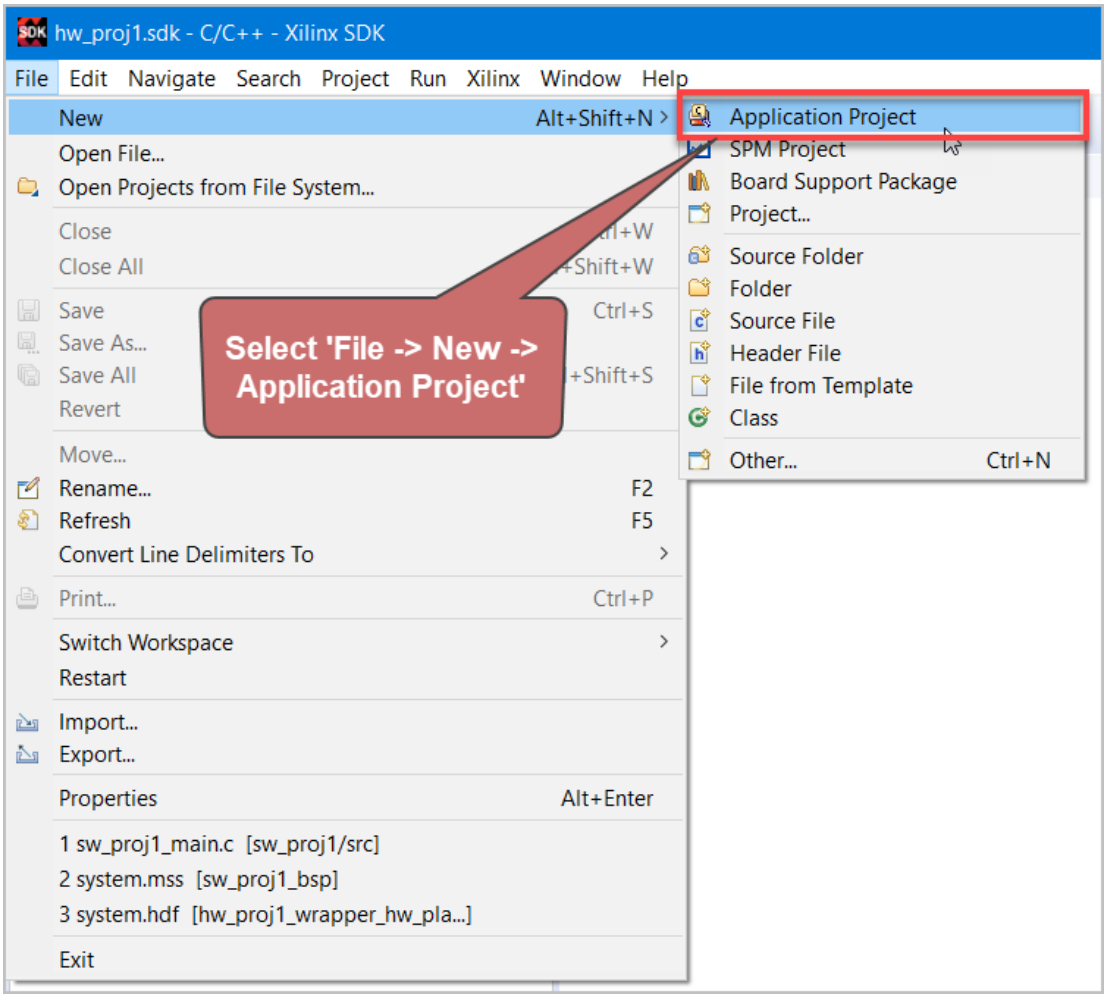


Figure 145. Select File->New-> Application Project

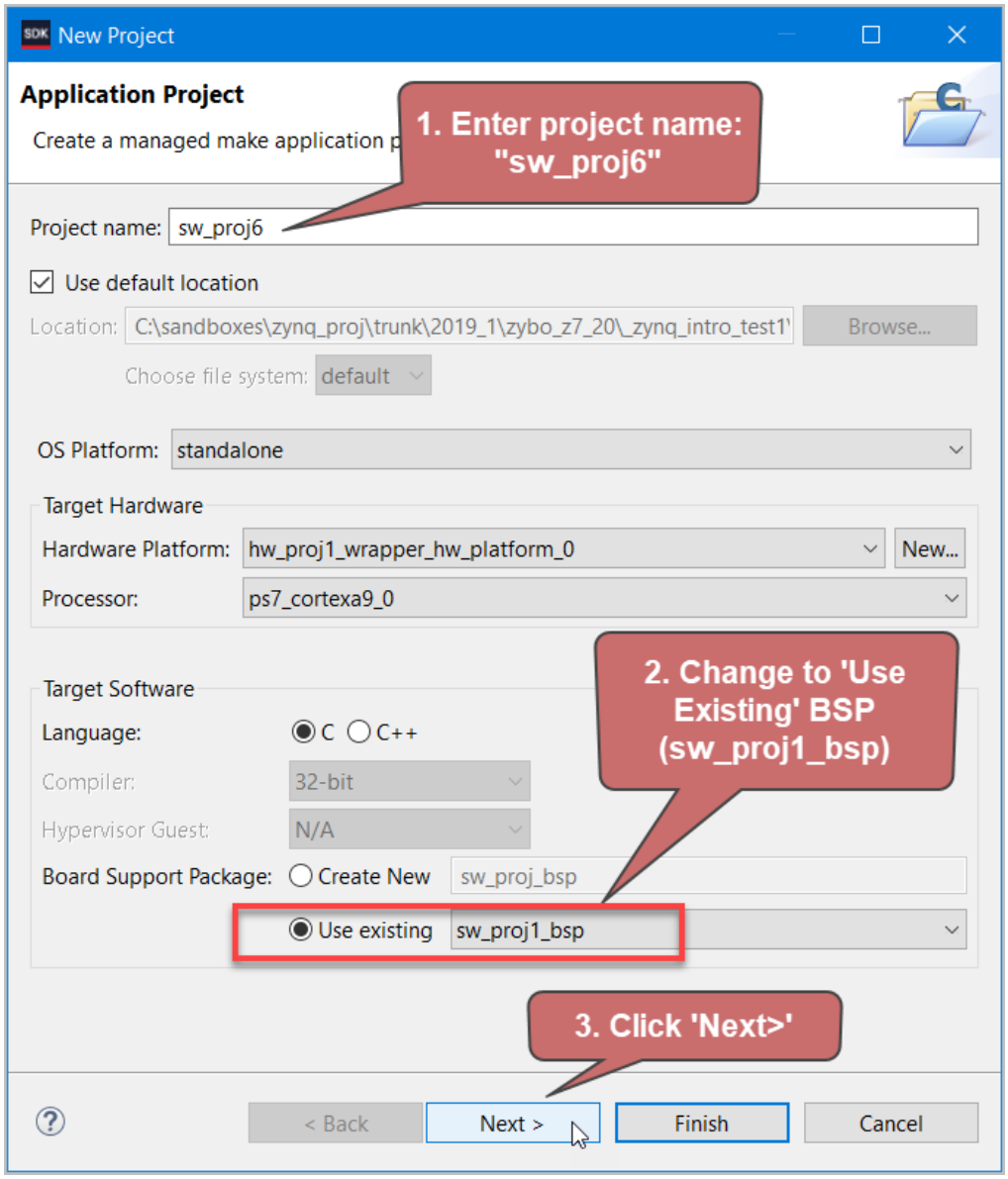


Figure 146. Enter the project details (part 1)

1. Enter the project name: sw_proj6
2. Board Support Package: Use existing
3. Click Next (don't click Finish!)

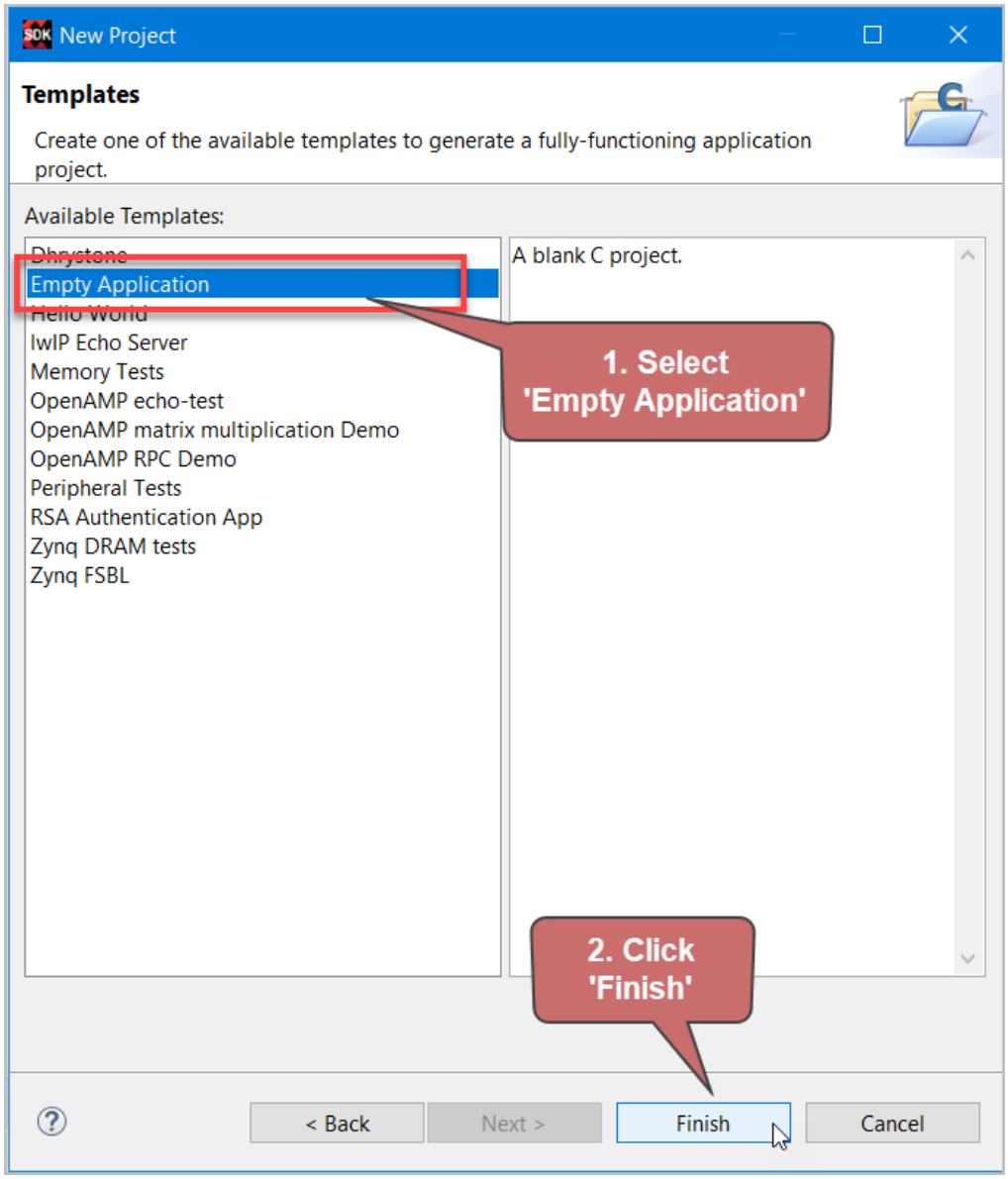


Figure 147. Enter the project details (part 2)

1. Select Empty Application
2. Click Finish

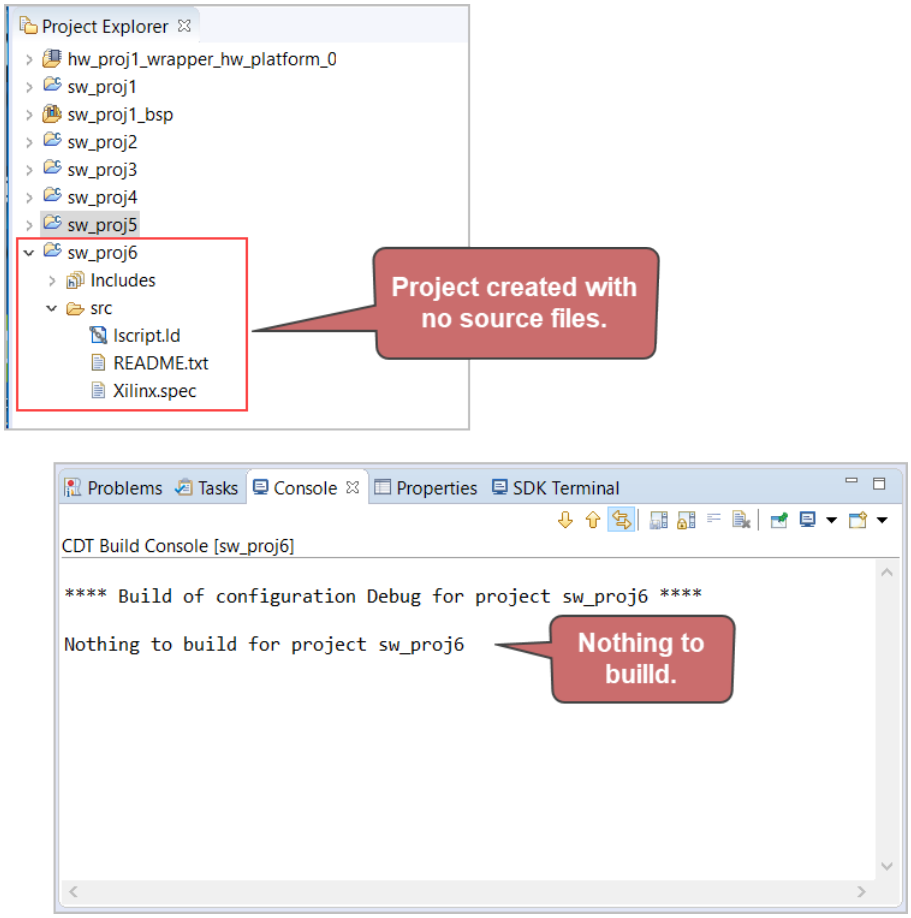


Figure 149. Project created with no source files

A project named “sw_proj6” is created with no project files. The CDT Build Console confirms that there is nothing to build.

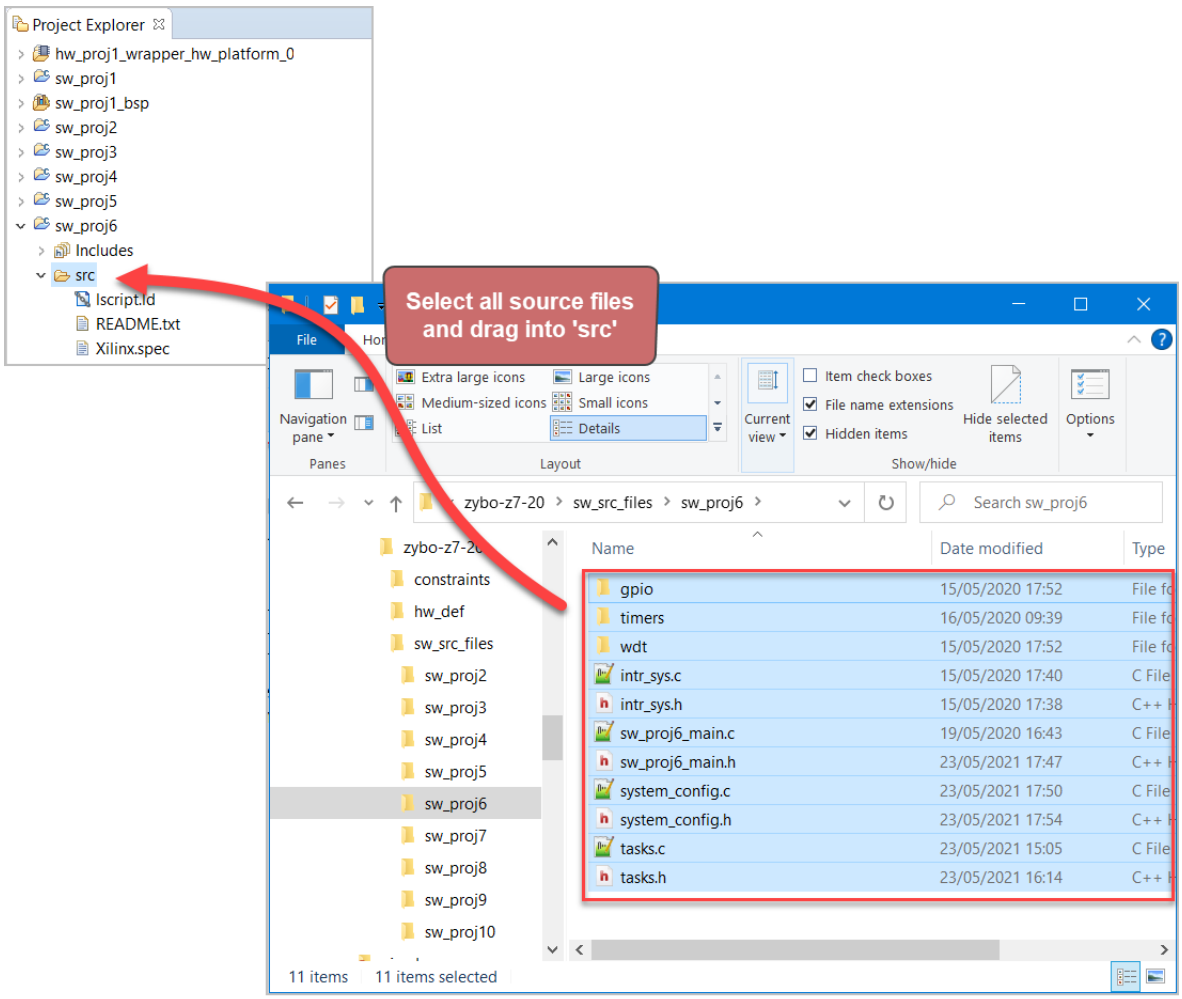


Figure 150. Drag the source files from Windows Explorer into the SDK project

Open the location where the source files for software project 6 are saved on your PC. Drag all the files and directory and from Windows explorer directly into the "**src**" folder in SDK.

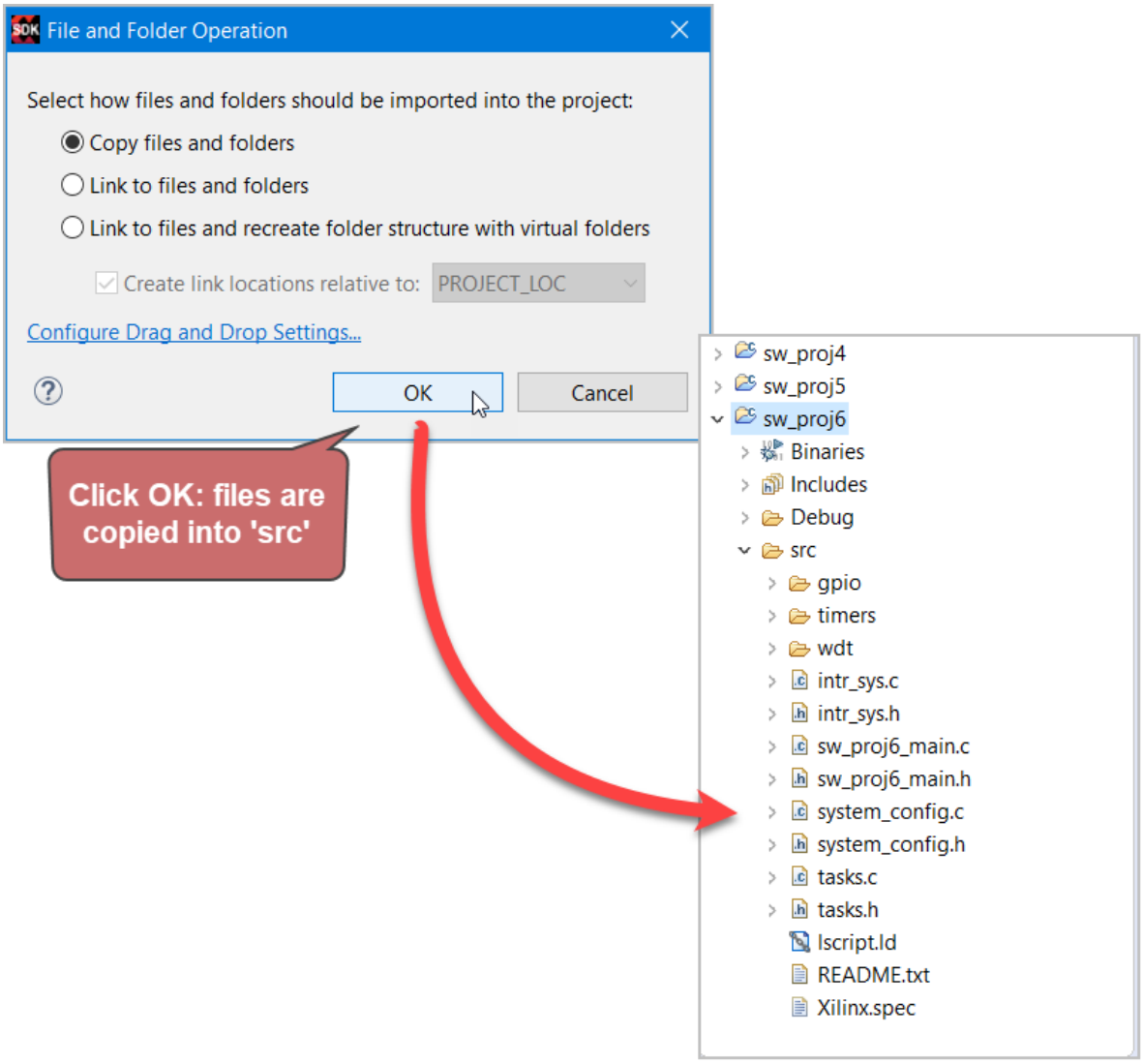


Figure 151. Click OK, and the files will be copied into the project.

In the File Operation dialog box, ensure “Copy files and folders” is checked, and click OK.

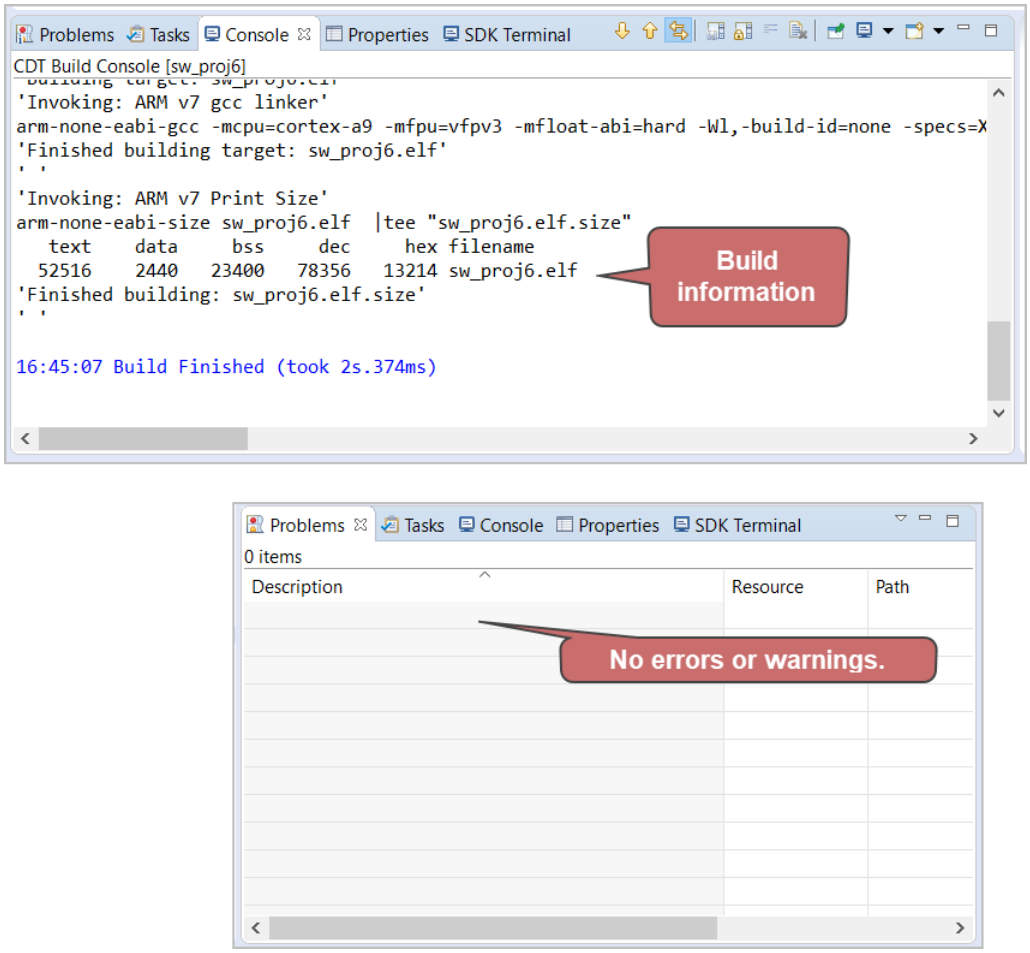


Figure 152. The project should build successfully

If "Build Automatically" is selected, the project should build successfully, with no errors or warnings.

[Exceptions: SDK 2018.3/SDK 2019.1 may indicate a warning as follows, which can be ignored:

#pragma message: For the sleep routines, Global timer is being used

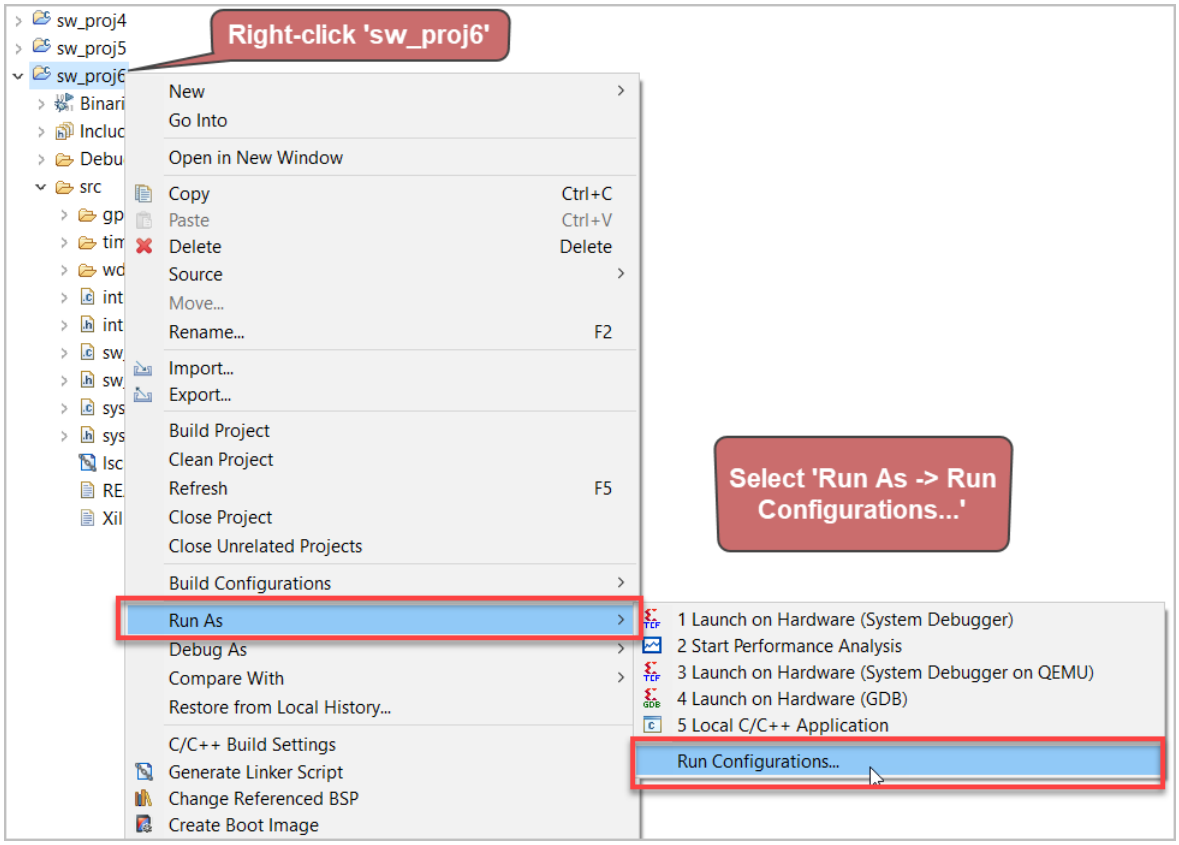


Figure 153. Right-click the software project and create a Run Configuration

To run the program, the first step is to create a Run configuration. Right-click on the project, and select the 'Run Configurations...' option.

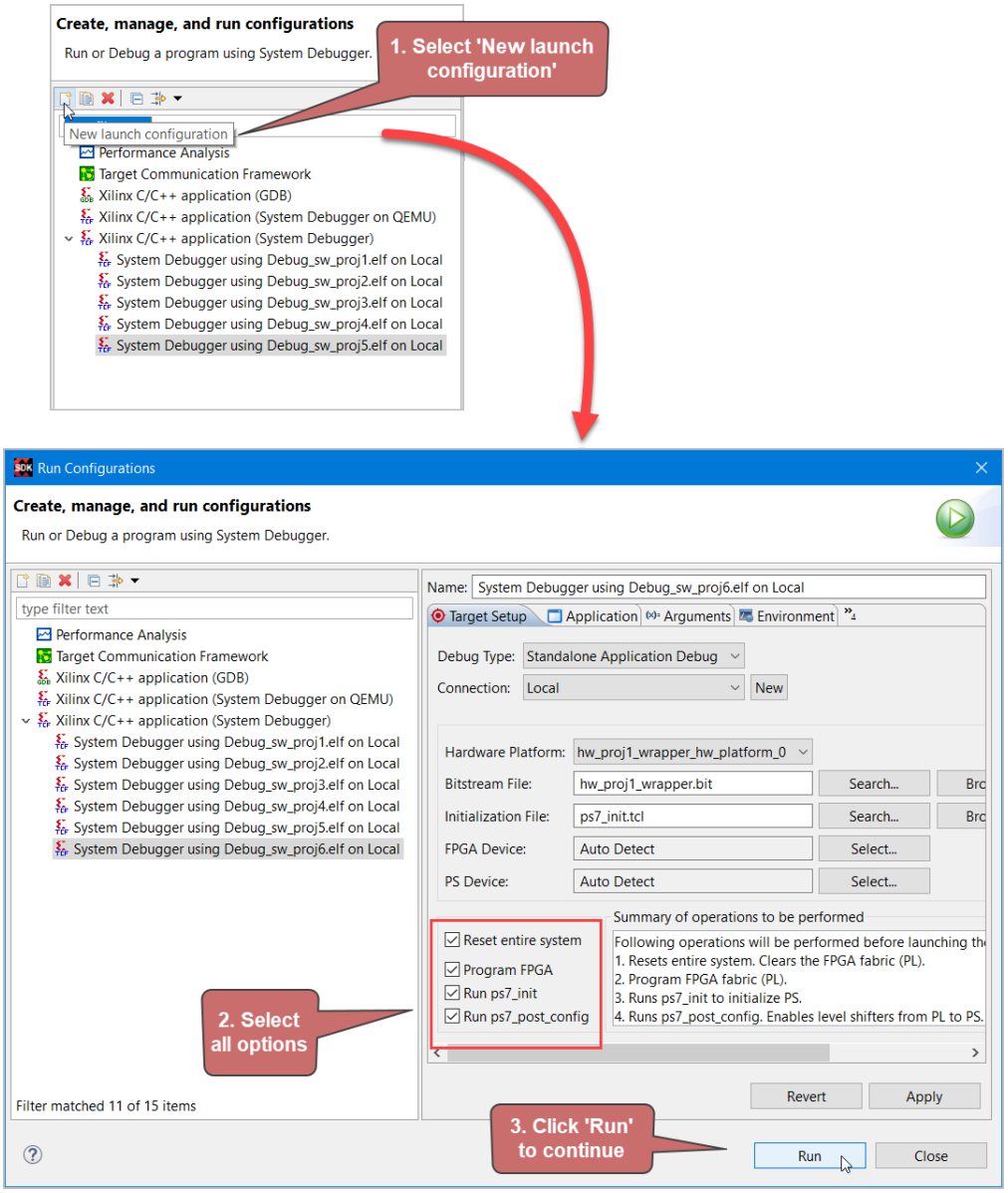


Figure 154. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

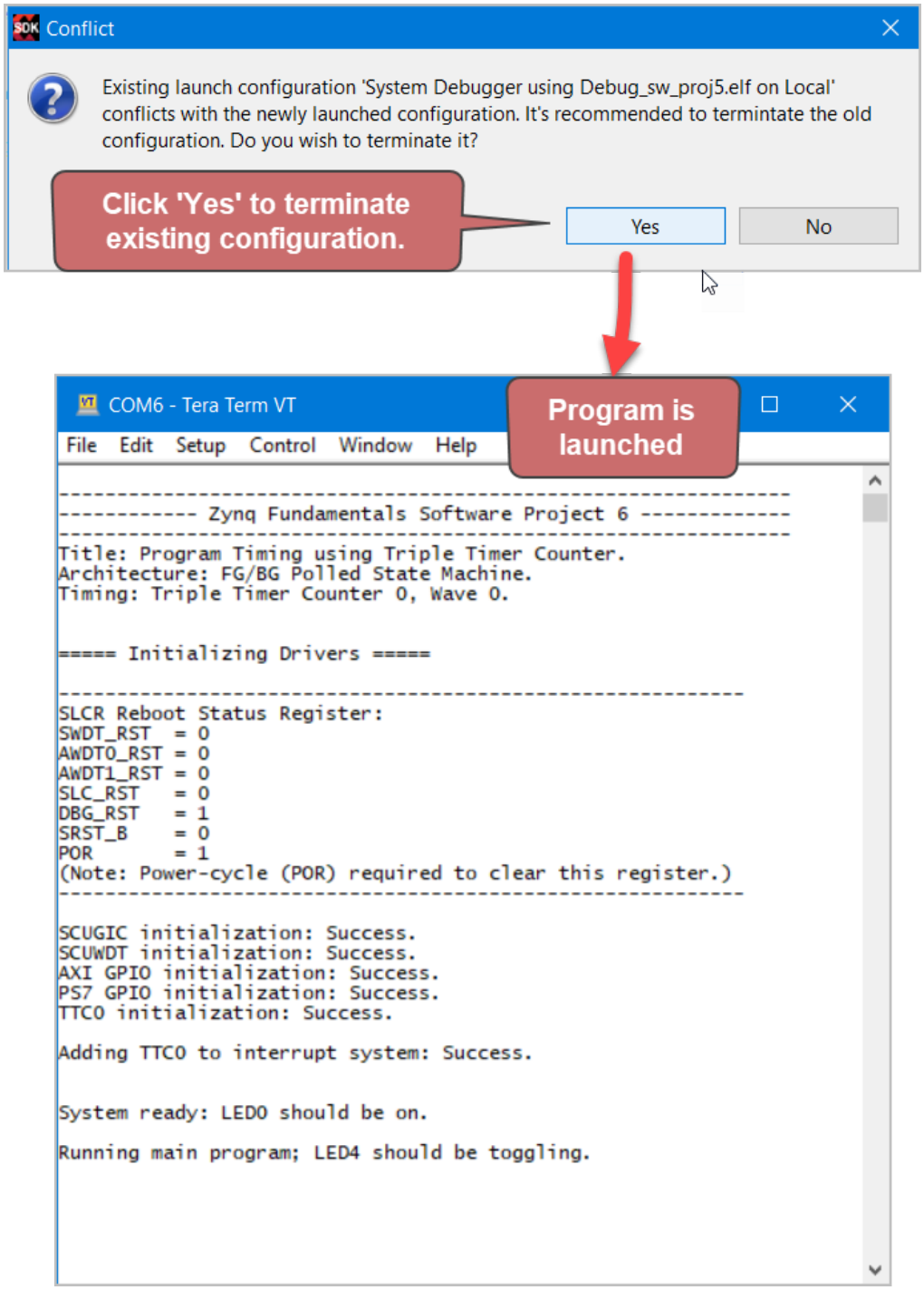


Figure 155. Terminal output for Software Project 6

If prompted, select 'Yes' to terminate any existing configuration. The FPGA will be programmed and the application will be downloaded to the processor. The terminal program should display the results for launching Software Project 6.

3.7 Software Project 7: Button De-bouncing

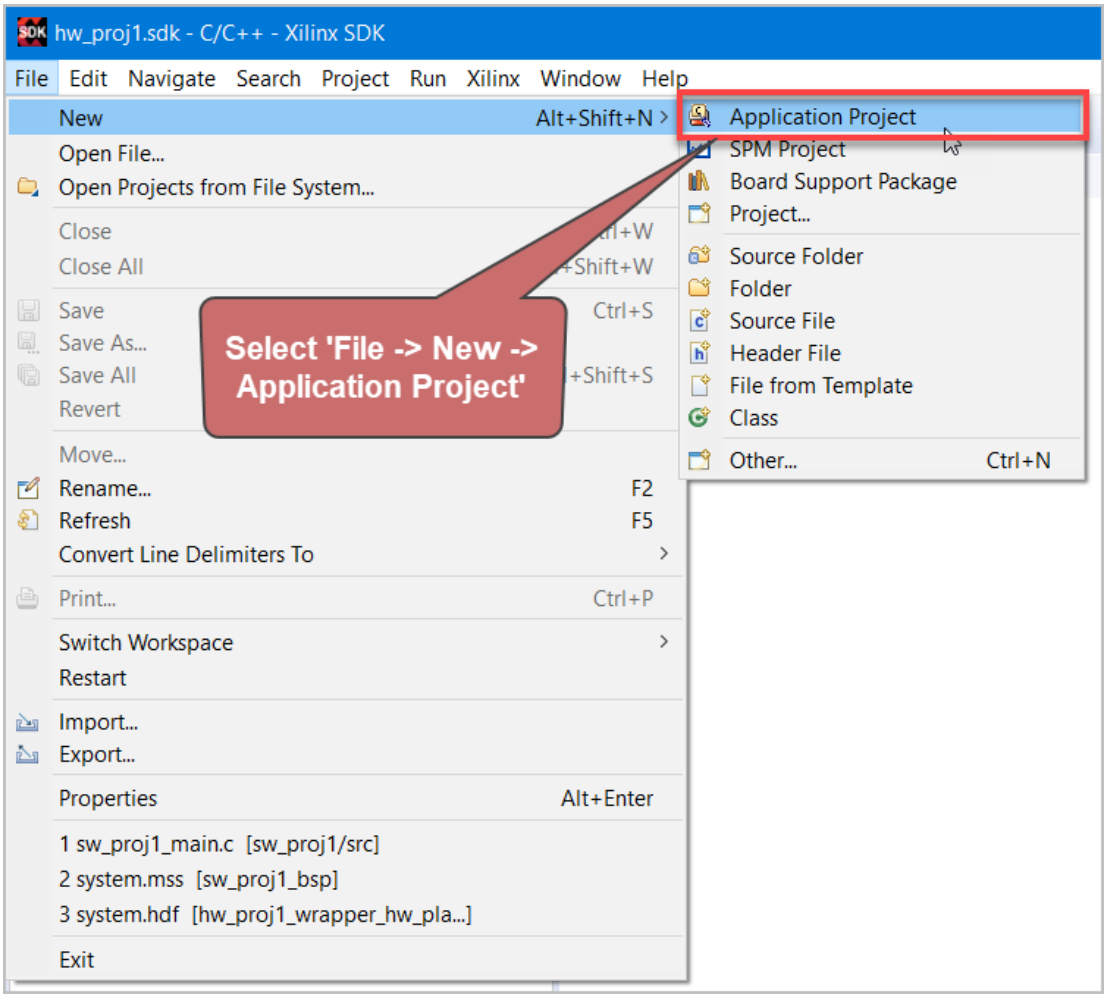


Figure 156. Select File->New-> Application Project

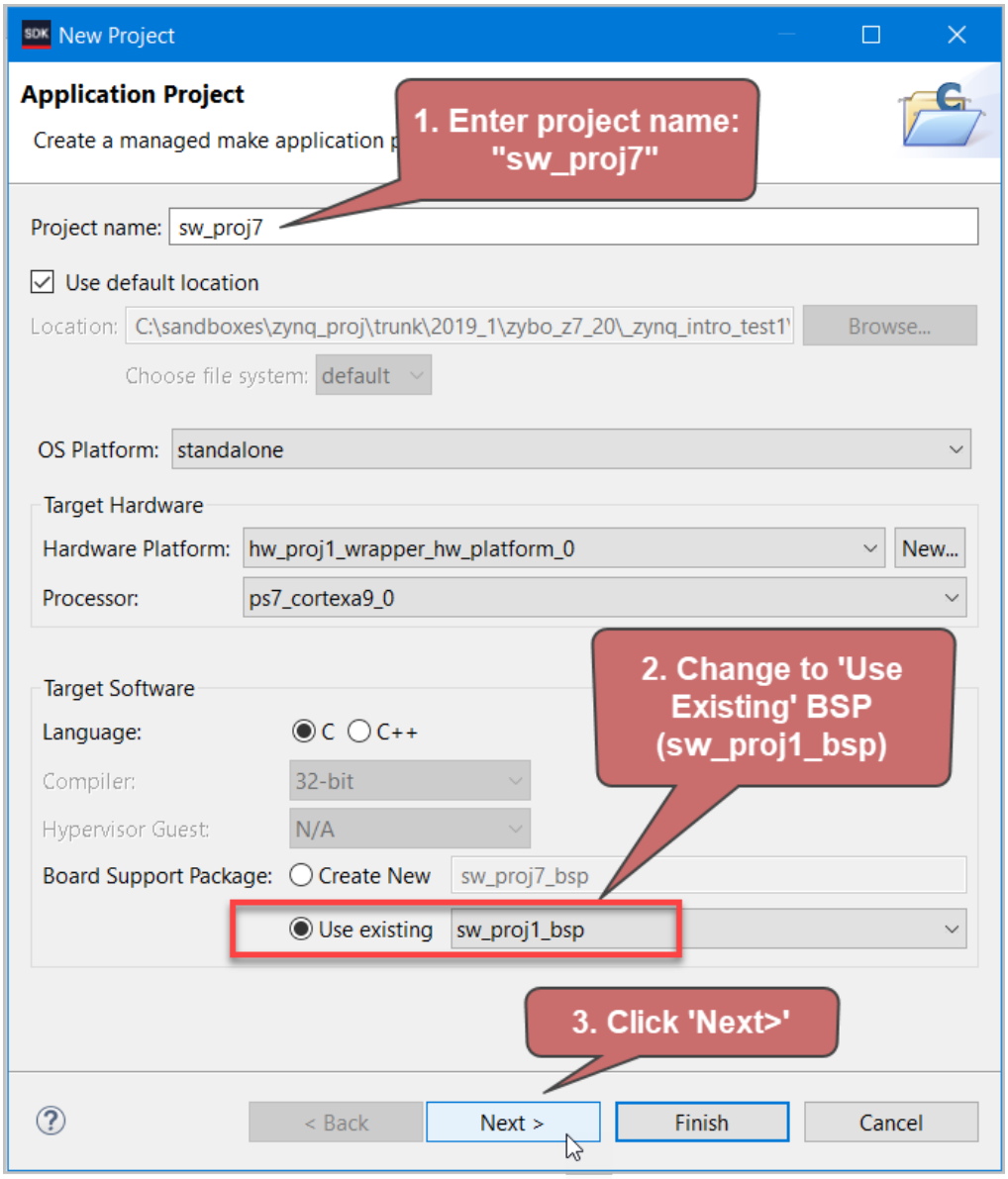


Figure 157. Enter the project details (part 1)

1. Enter the project name: sw_proj7
2. Board Support Package: Use existing
3. Click Next (don't click Finish!)

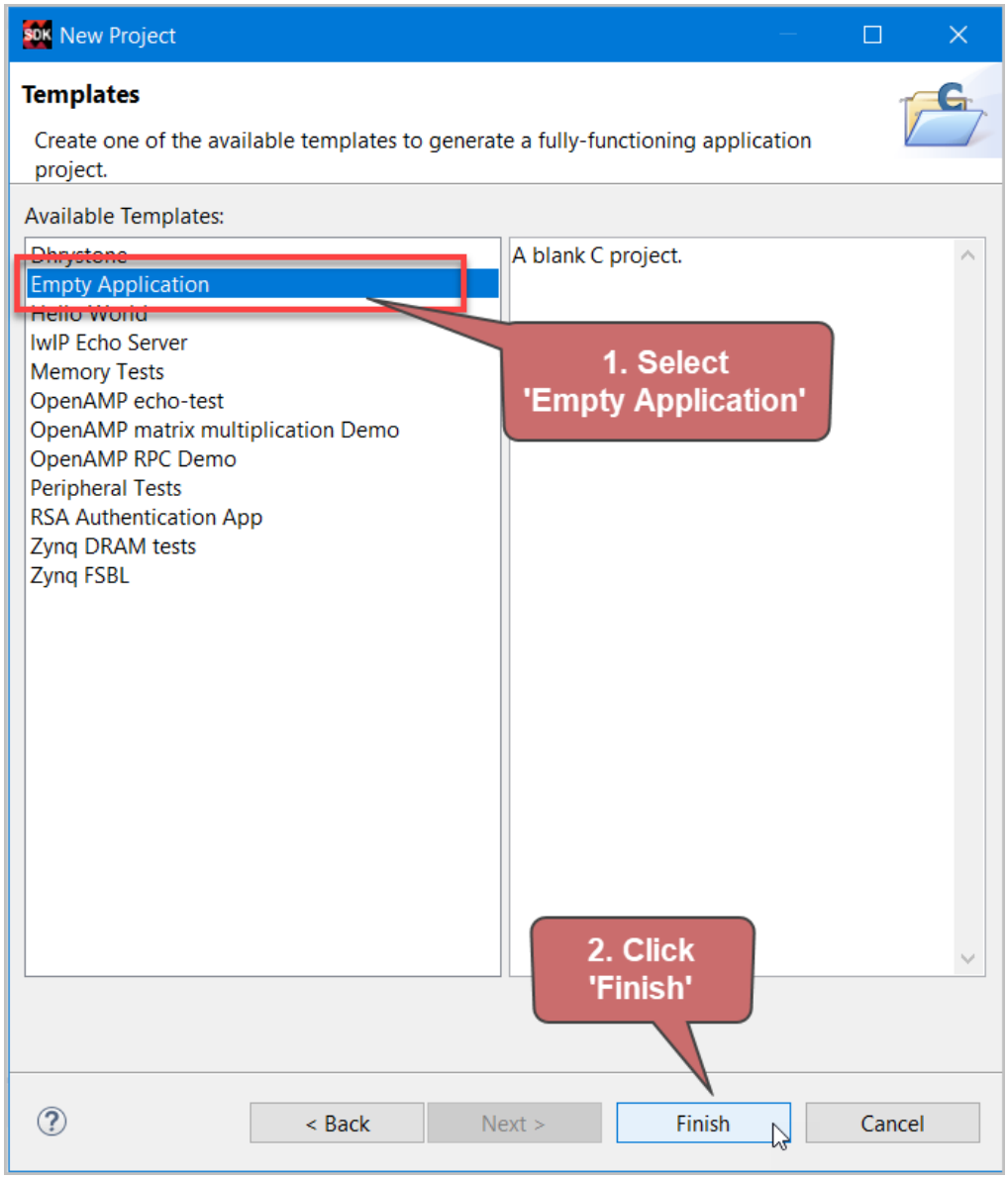


Figure 158. Enter the project details (part 2)

1. Select Empty Application
2. Click Finish

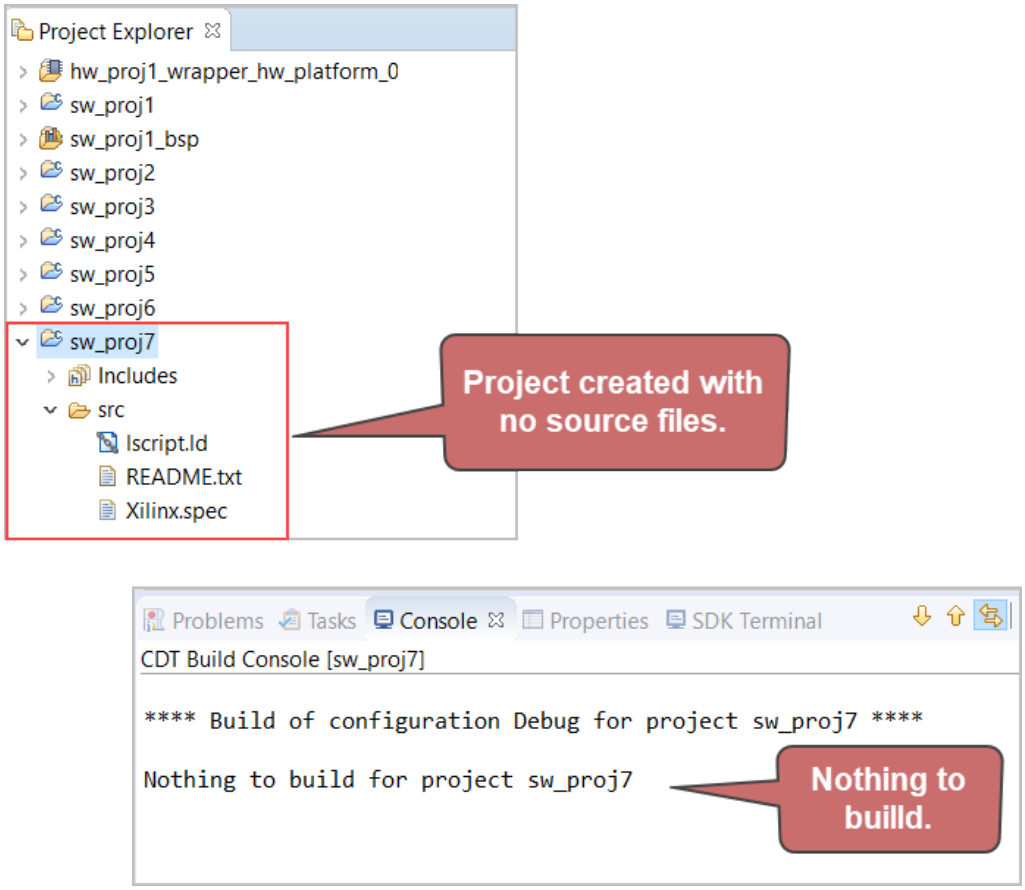


Figure 160. Project created with no source files

A project named “sw_proj7” is created with no project files. The CDT Build Console confirms that there is nothing to build.

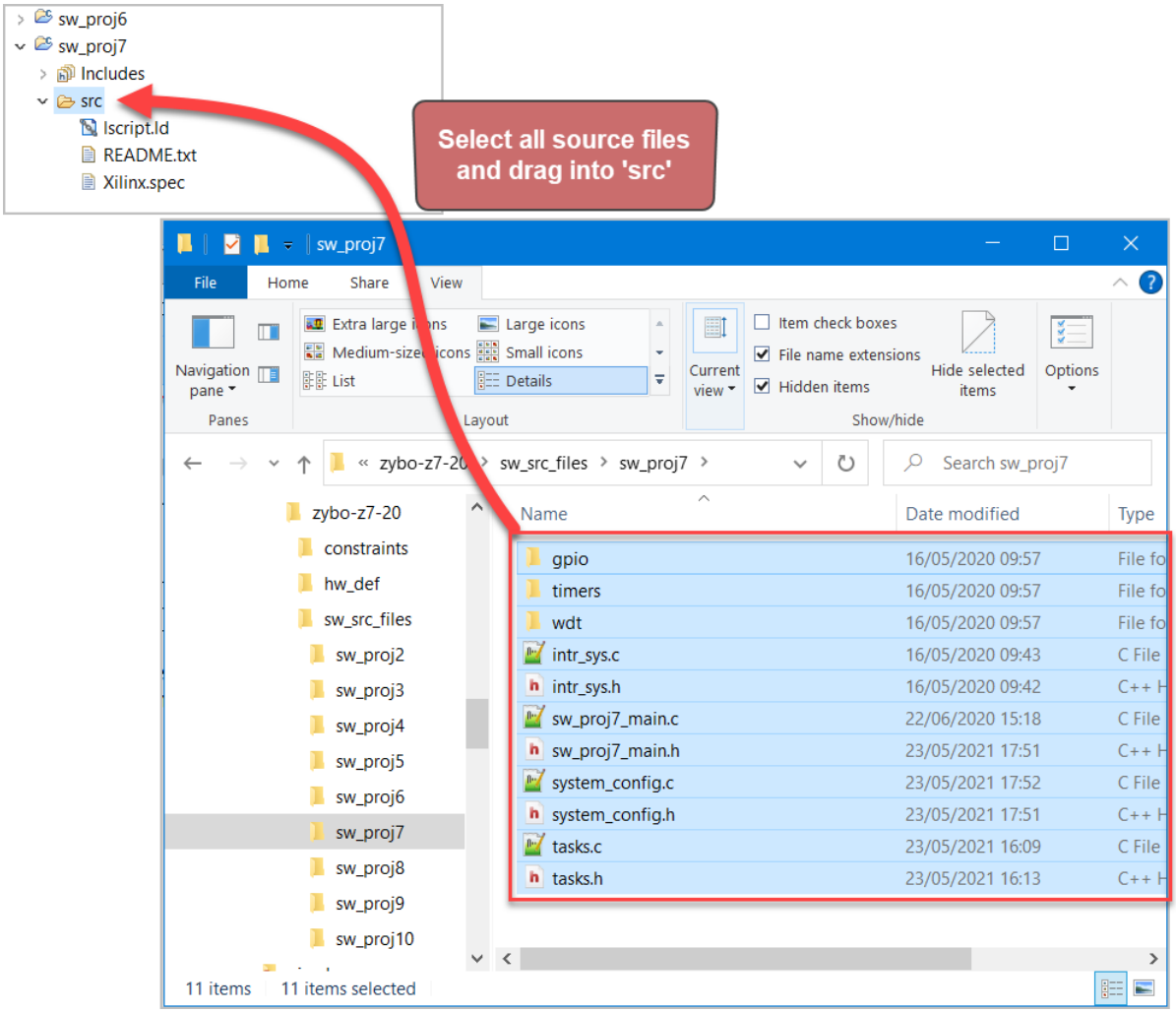


Figure 161. Drag the source files from Windows Explorer into the SDK project

Open the location where the source files for software project 7 are saved on your PC. Drag all the files and directory and from Windows explorer directly into the "**src**" folder in SDK.

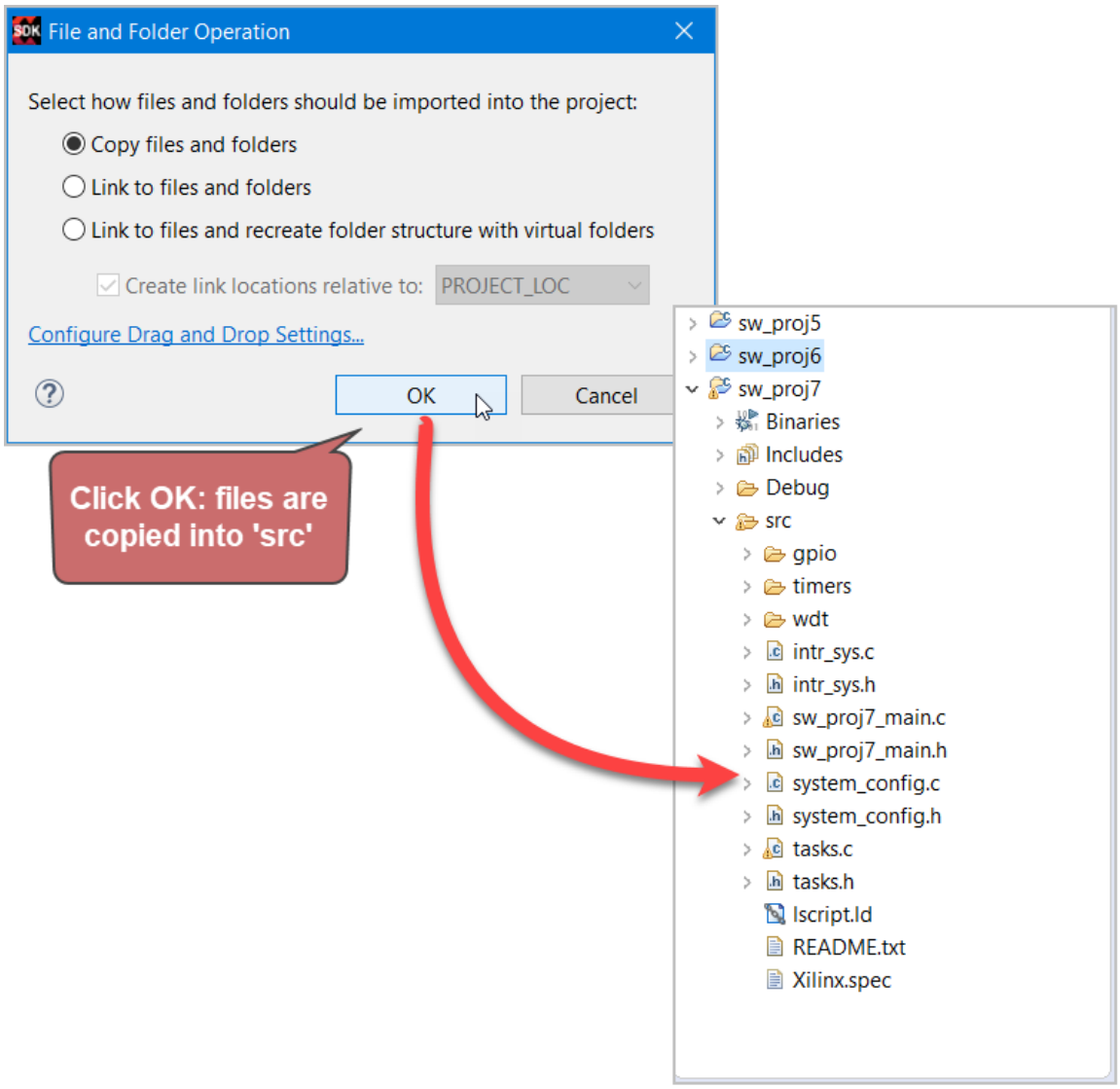


Figure 162. Click OK, and the files will be copied into the project.

In the File Operation dialog box, ensure "Copy files and folders" is checked, and click OK.

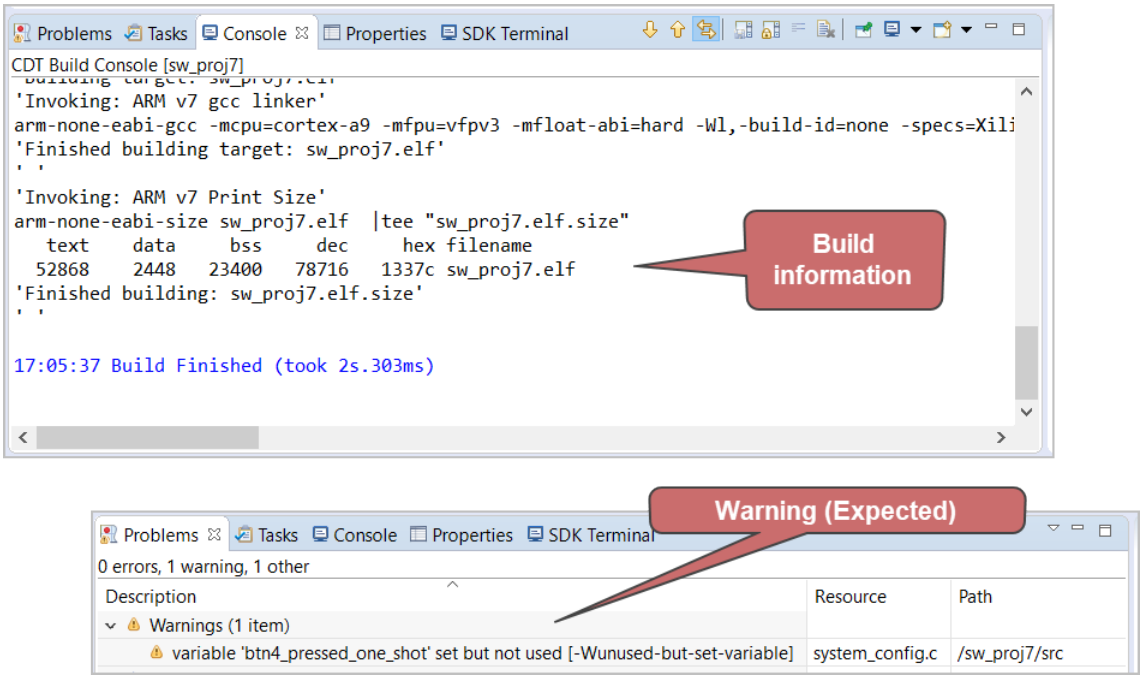


Figure 163. The project should build successfully, but a warning will be generated (this is expected).

If "Build Automatically" is selected, the project should build successfully, although for this project, a warning is expected, as follows:

- variable 'btn4_pressed_one_shot' set but not used

[SDK 2018.3/SDK 2019.1 may also indicate an information message as follows, which can be ignored:

#pragma message: For the sleep routines, Global timer is being used

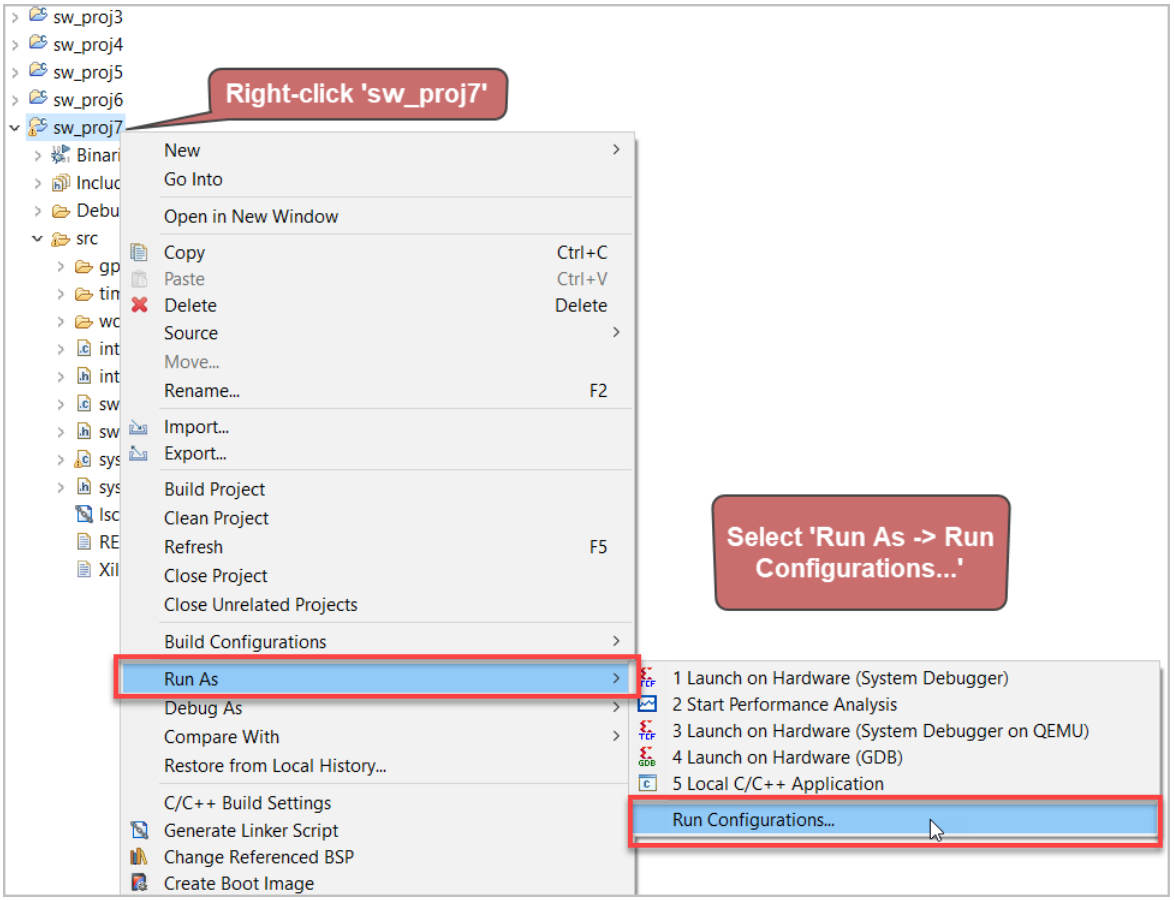


Figure 164. Right-click the software project and create a Run Configuration

To run the program, the first step is to create a Run configuration. Right-click on the project, and select the 'Run Configurations...' option.

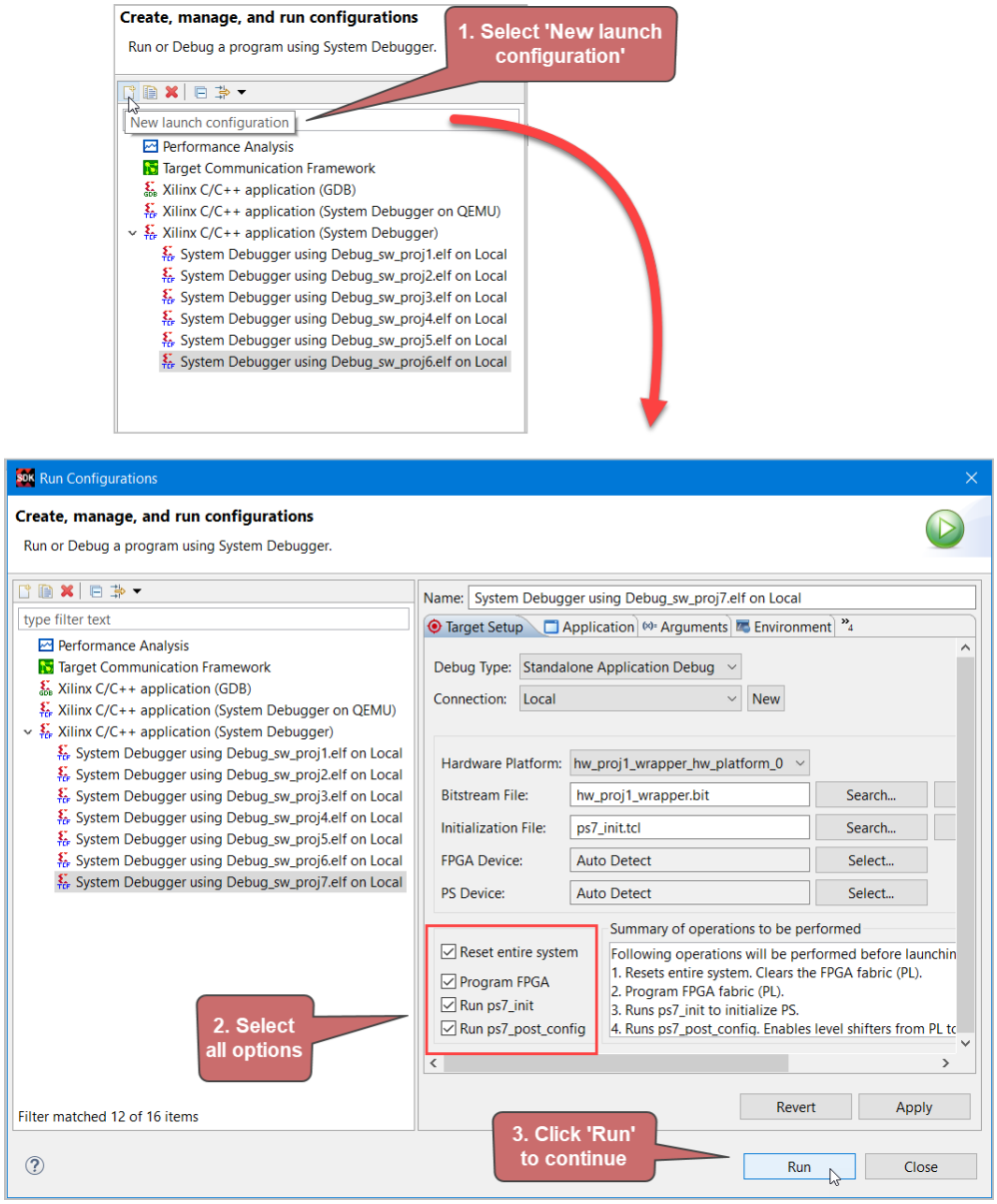


Figure 165. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

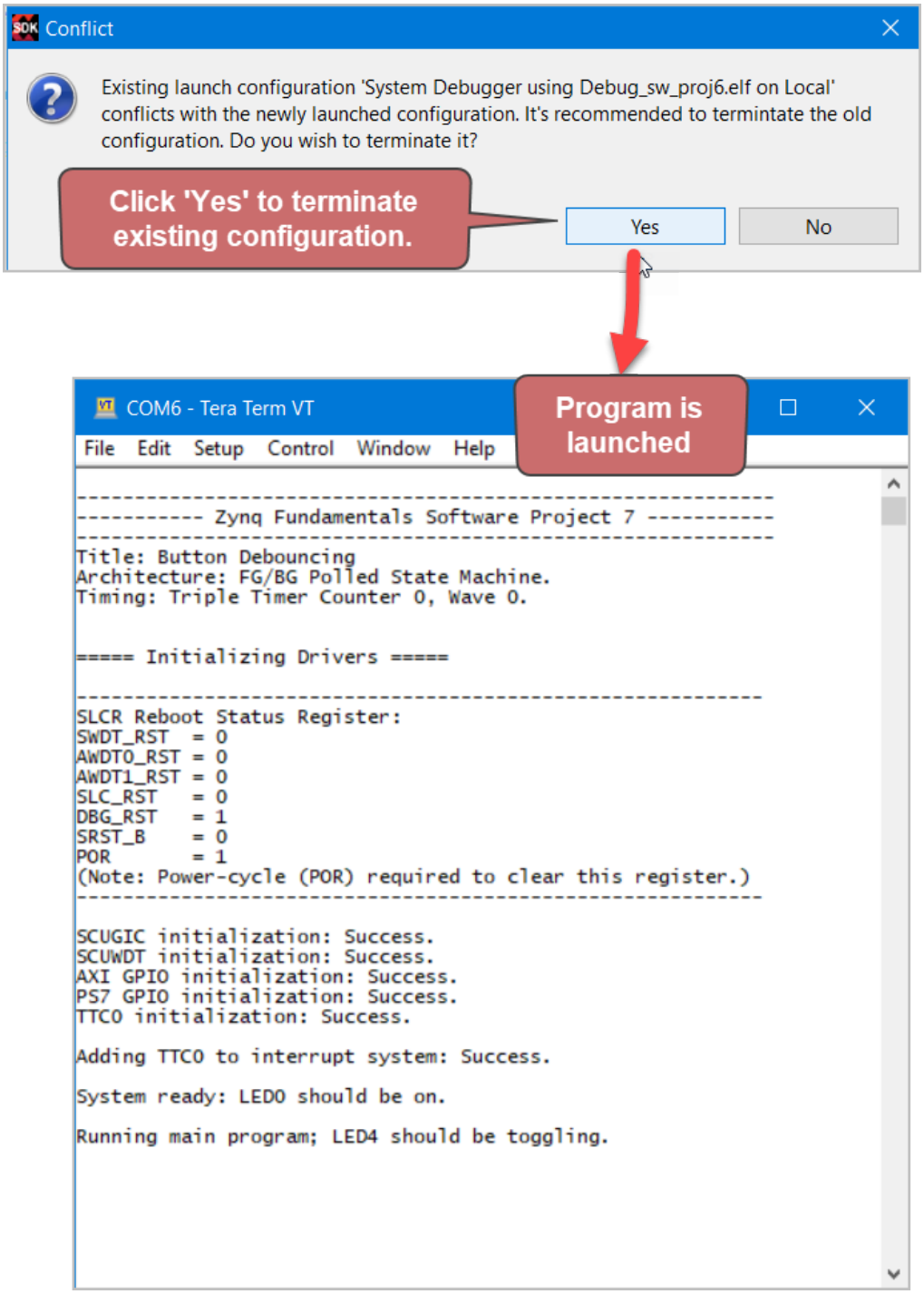


Figure 166. Terminal output for Software Project 7

If prompted, select 'Yes' to terminate any existing configuration. The FPGA will be programmed and the application will be downloaded to the processor. The terminal program should display the results for launching Software Project 7.

3.8 Software Project 8: UART Command Handler

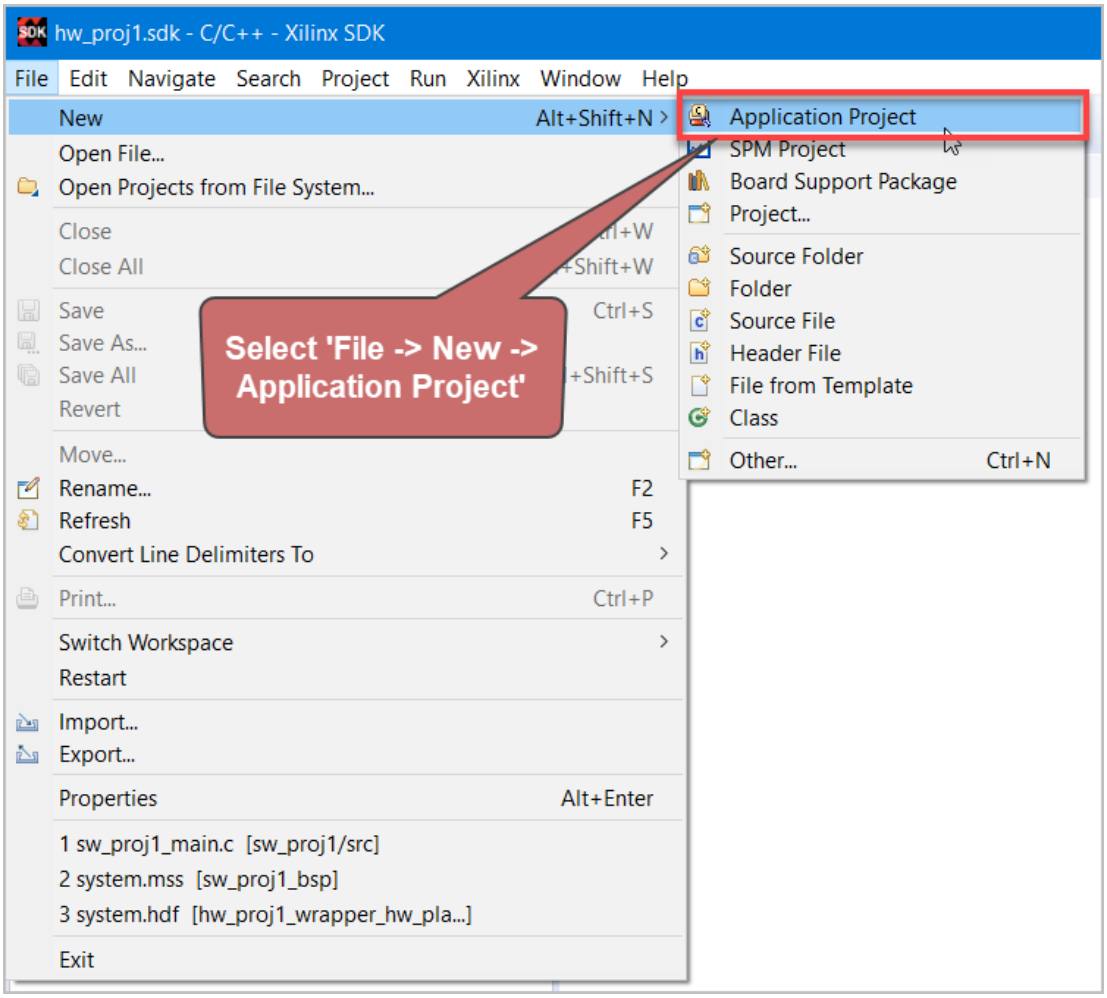


Figure 167. Select File->New-> Application Project

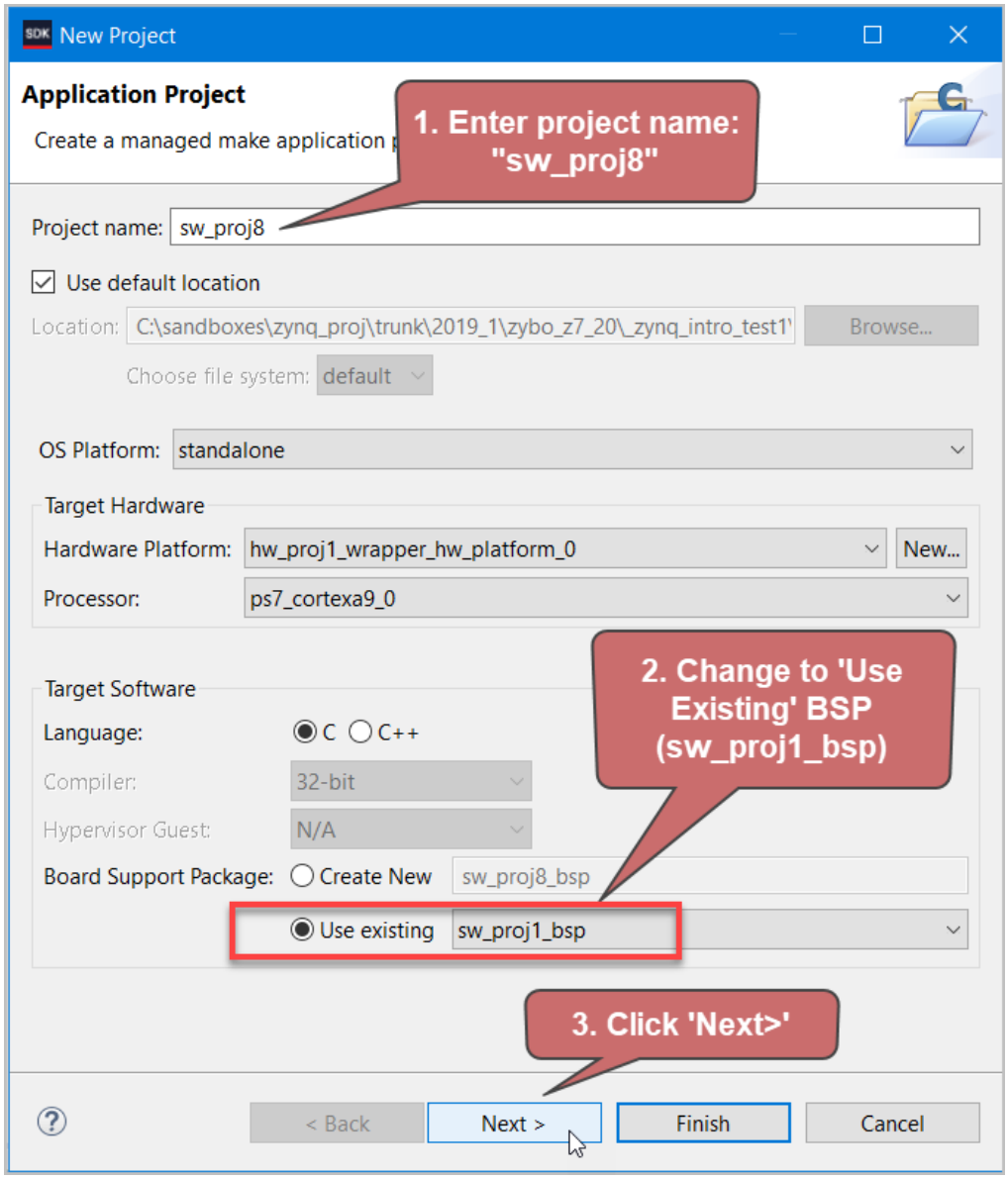


Figure 168. Enter the project details (part 1)

1. Enter the project name: sw_proj8
2. Board Support Package: Use existing
3. Click Next (don't click Finish!)

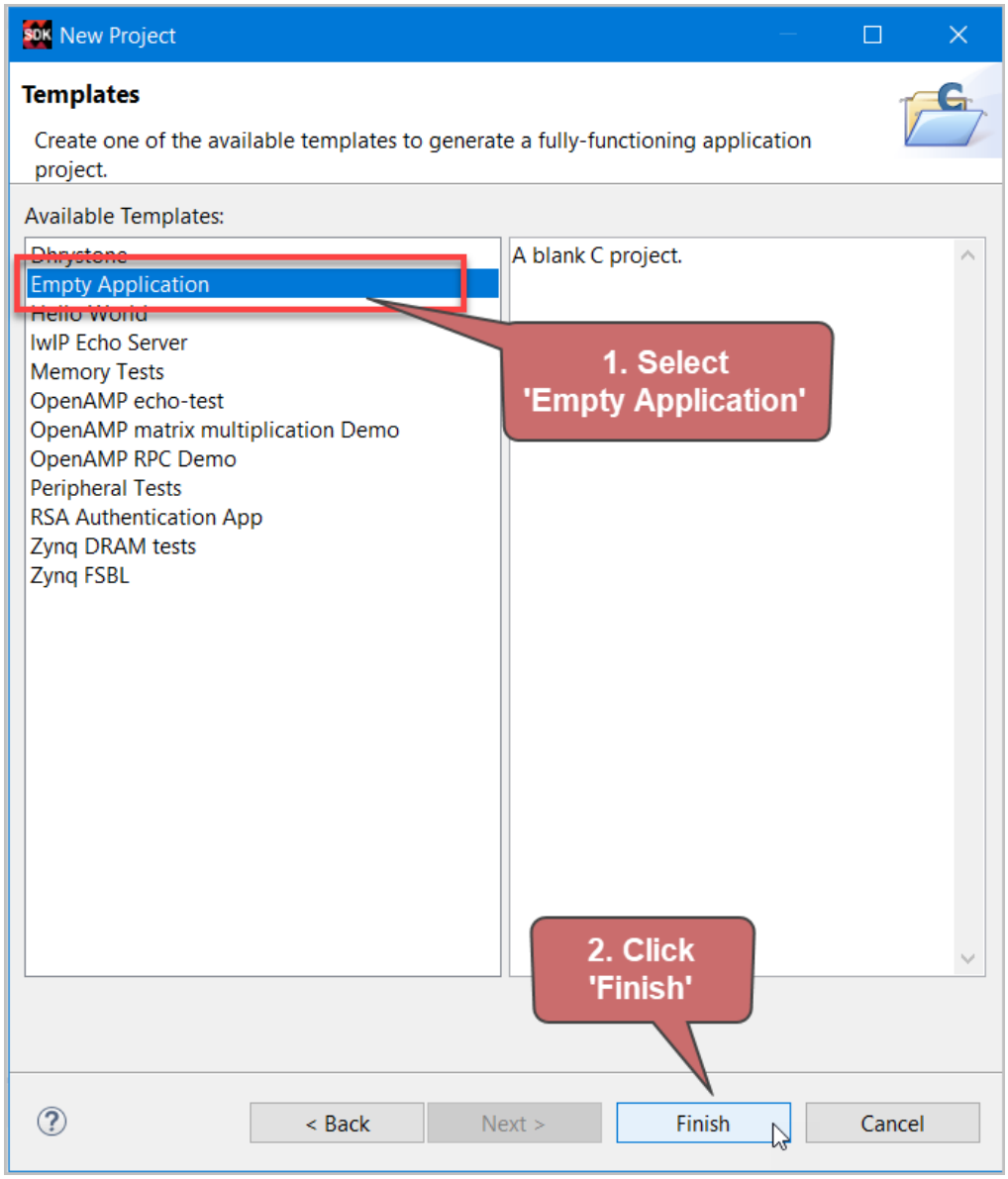


Figure 169. Enter the project details (part 2)

1. Select Empty Application
2. Click Finish

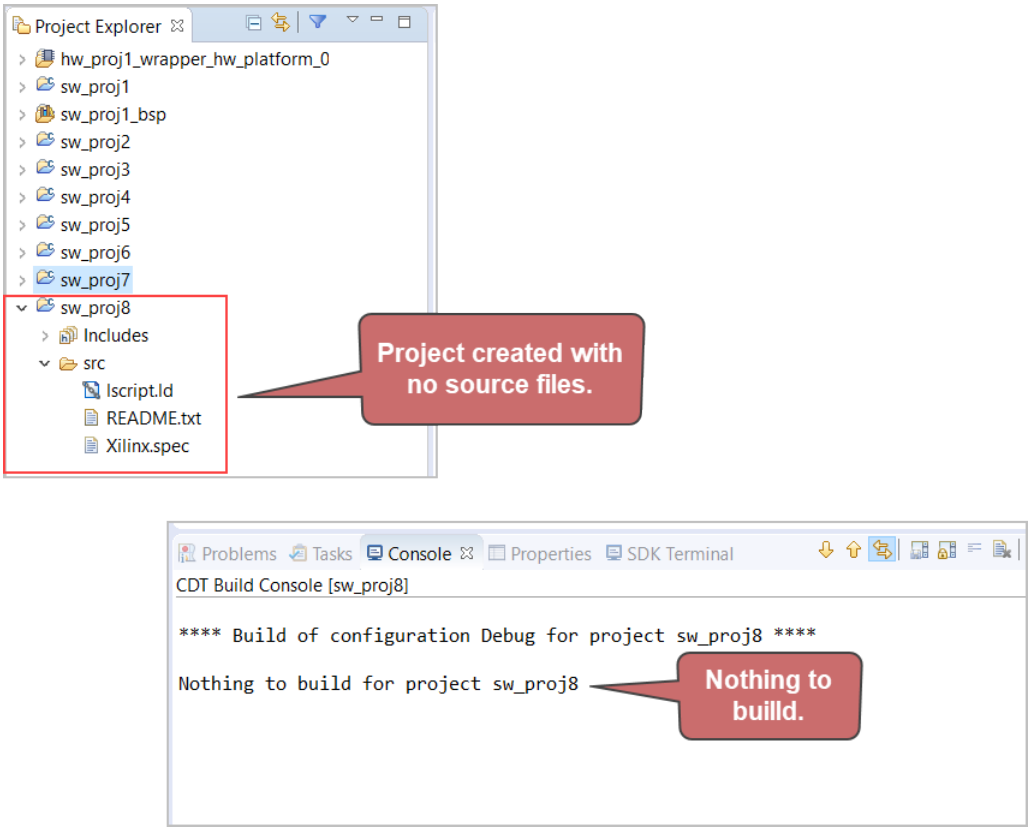


Figure 171. Project created with no source files

A project named “sw_proj8” is created with no project files. The CDT Build Console confirms that there is nothing to build.

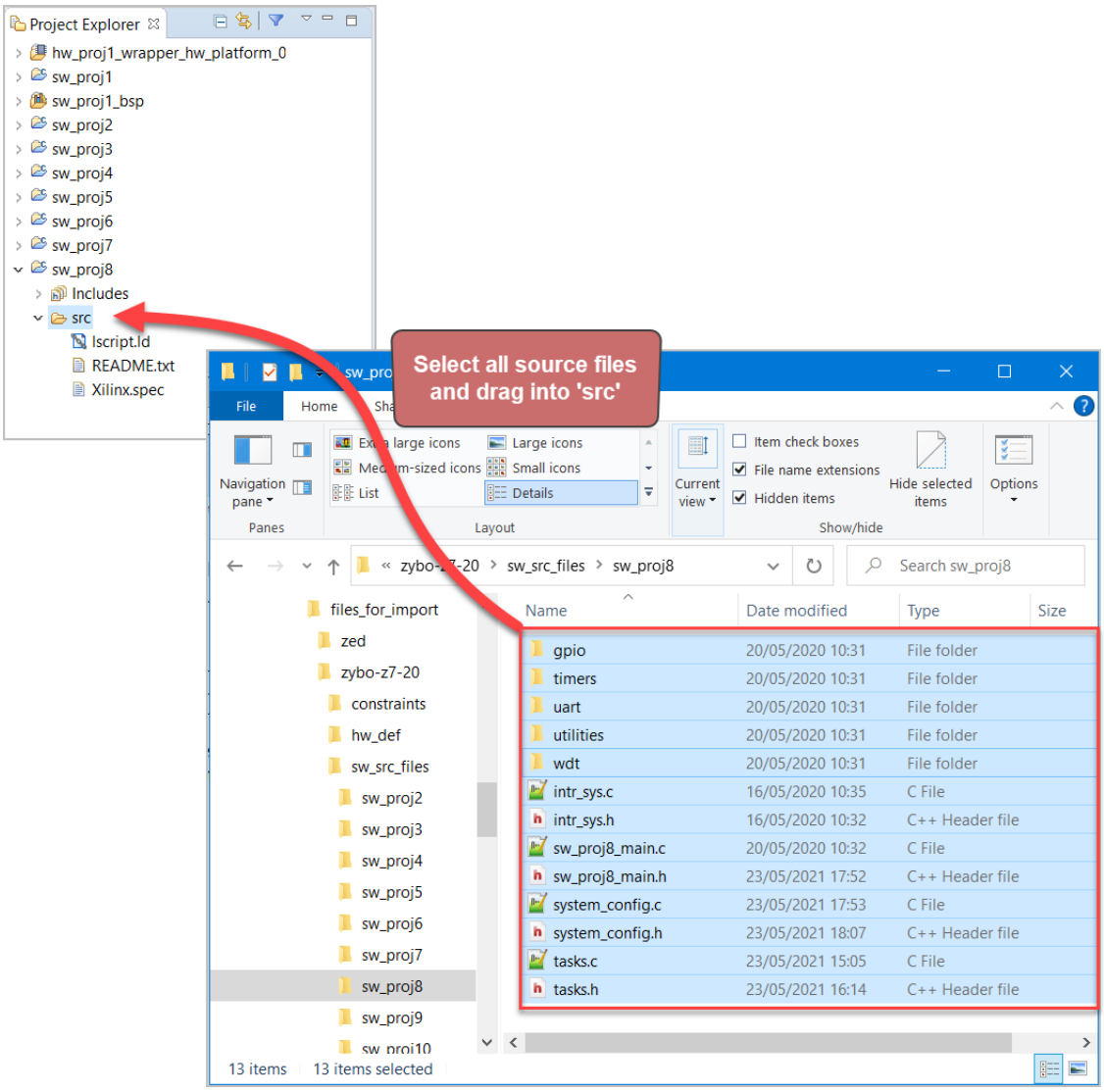


Figure 172. Drag the source files from Windows Explorer into the SDK project

Open the location where the source files for software project 8 are saved on your PC. Drag all the files and directory and from Windows explorer directly into the "**src**" folder in SDK.

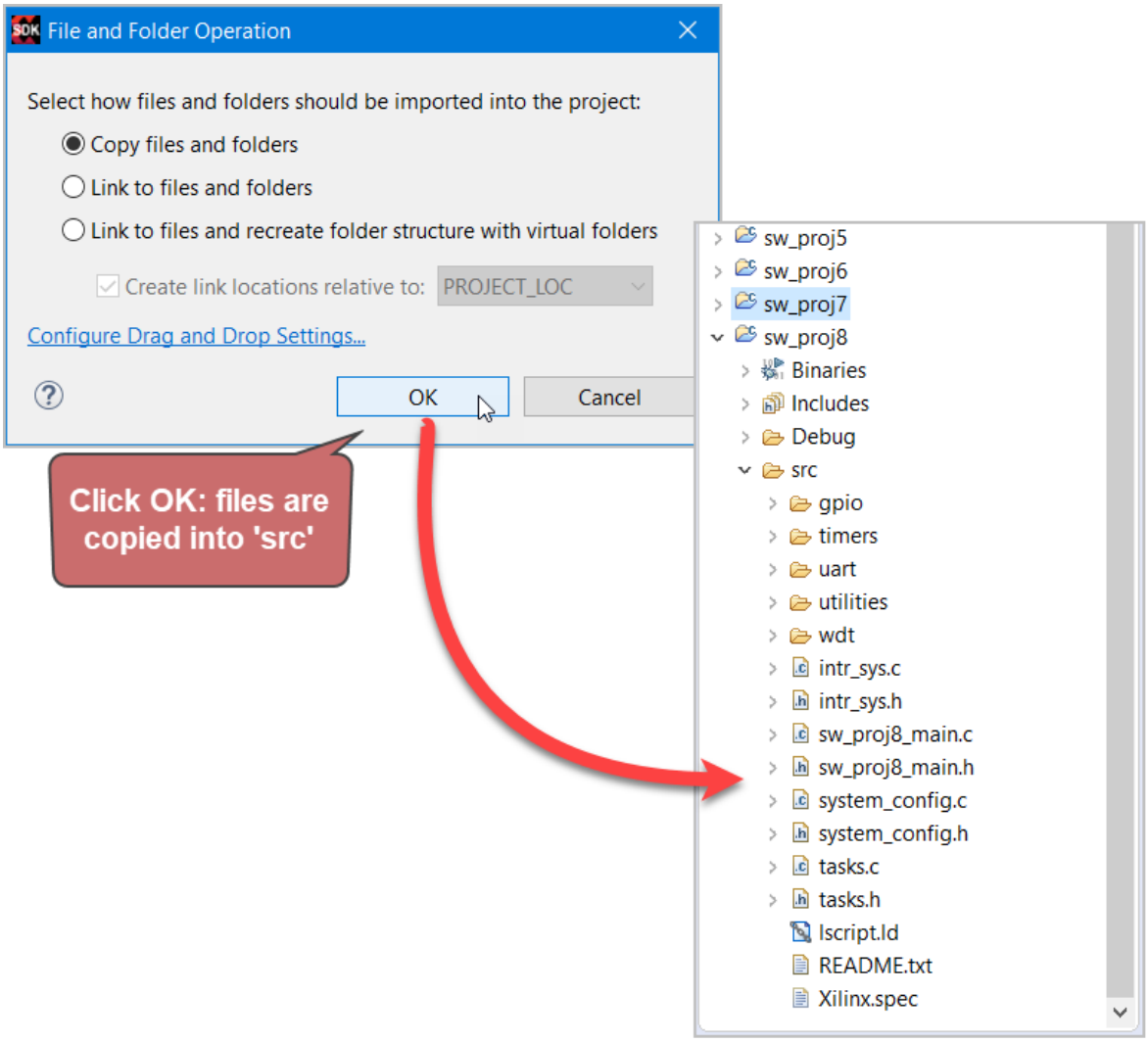


Figure 173. Click OK, and the files will be copied into the project.

In the File Operation dialog box, ensure “Copy files and folders” is checked, and click OK.

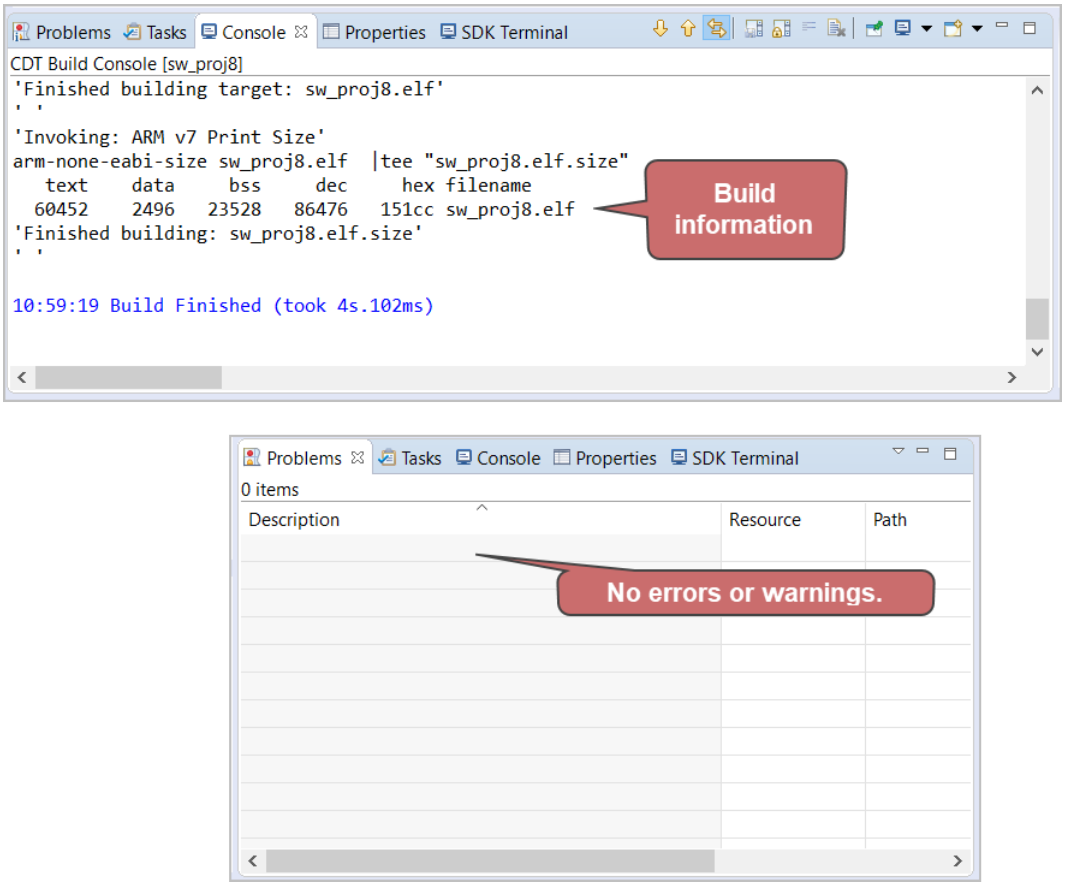


Figure 174. The project should build successfully.

If "Build Automatically" is selected, the project should build successfully, with no errors or warnings.

[Exceptions: SDK 2018.3/SDK 2019.1 may indicate a warning as follows, which can be ignored:

#pragma message: For the sleep routines, Global timer is being used

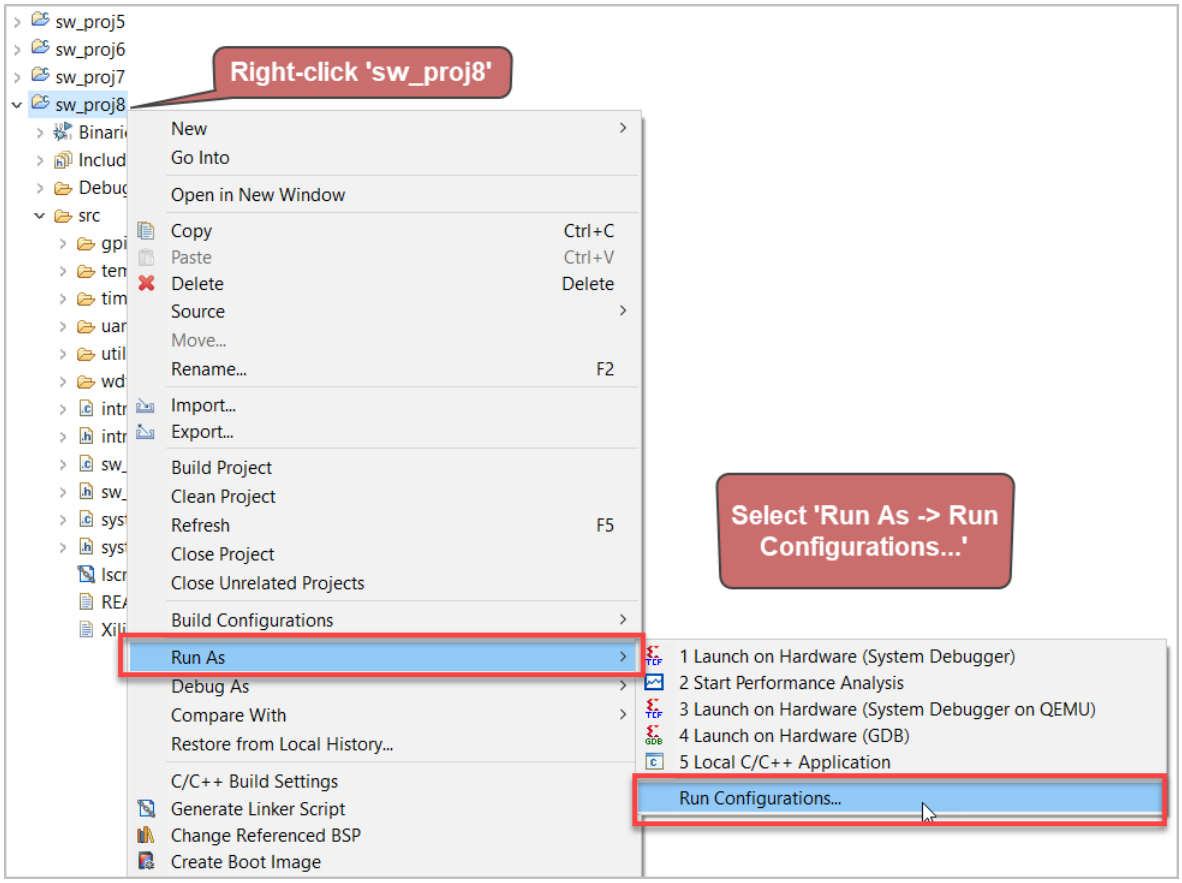


Figure 175. Right-click the software project and create a Run Configuration

To run the program, the first step is to create a Run configuration. Right-click on the project, and select the 'Run Configurations...' option.

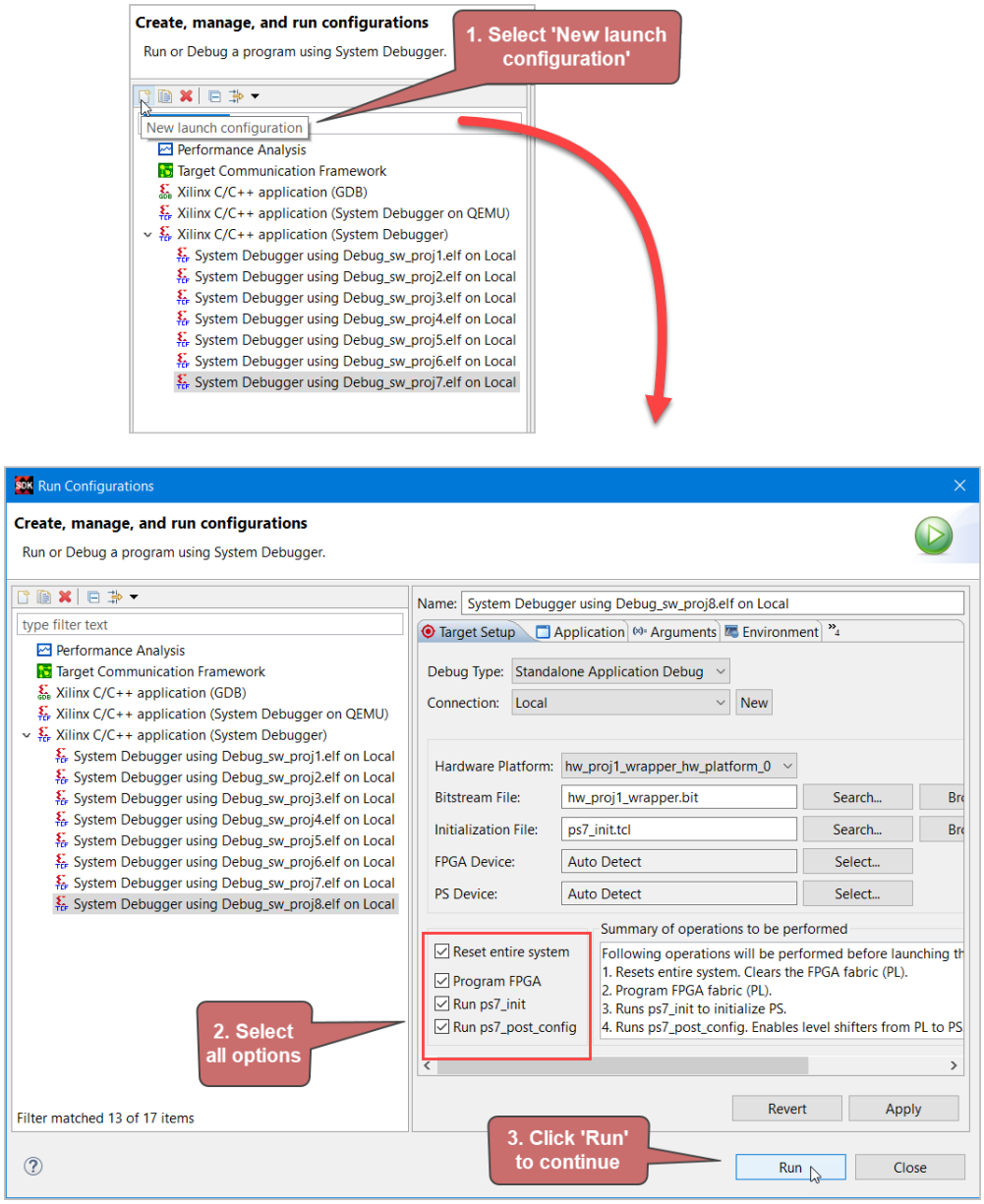


Figure 176. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

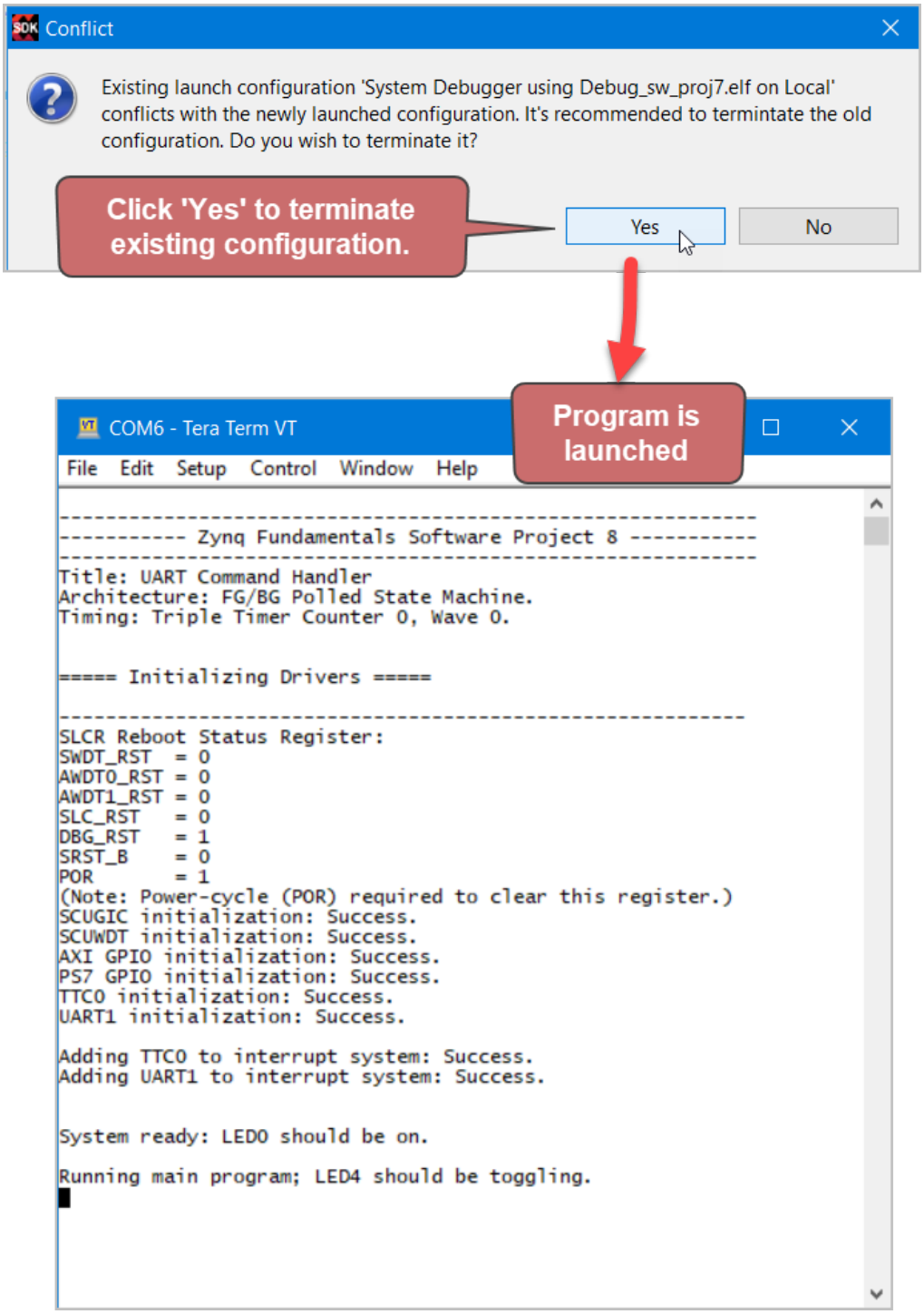


Figure 177. Terminal output for Software Project 8

If prompted, select 'Yes' to terminate any existing configuration. The FPGA will be programmed and the application will be downloaded to the processor. The terminal program should display the results for launching Software Project 8.

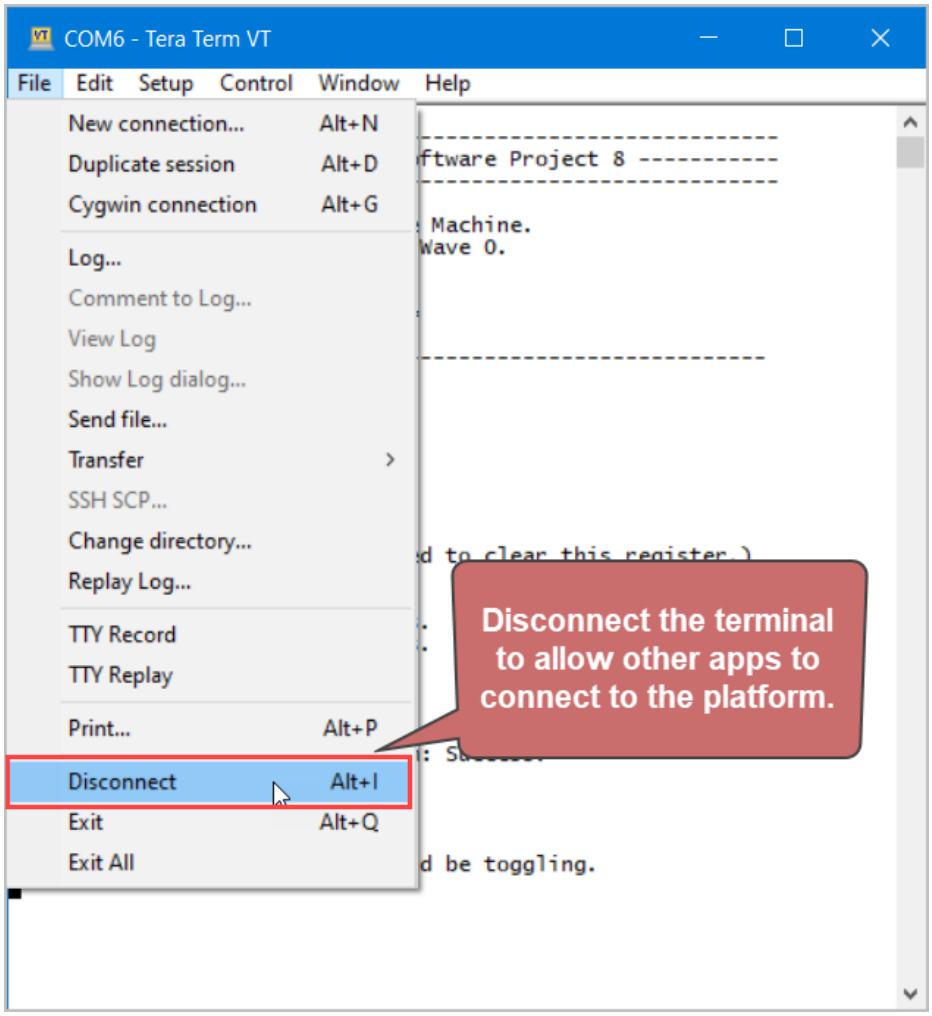


Figure 178. Disconnect the platform.

If the user wants to communicate with the command handler on the platform using a host application, the terminal needs to be disconnected.

3.9 Software Project 9: Programmable Logic Interrupts

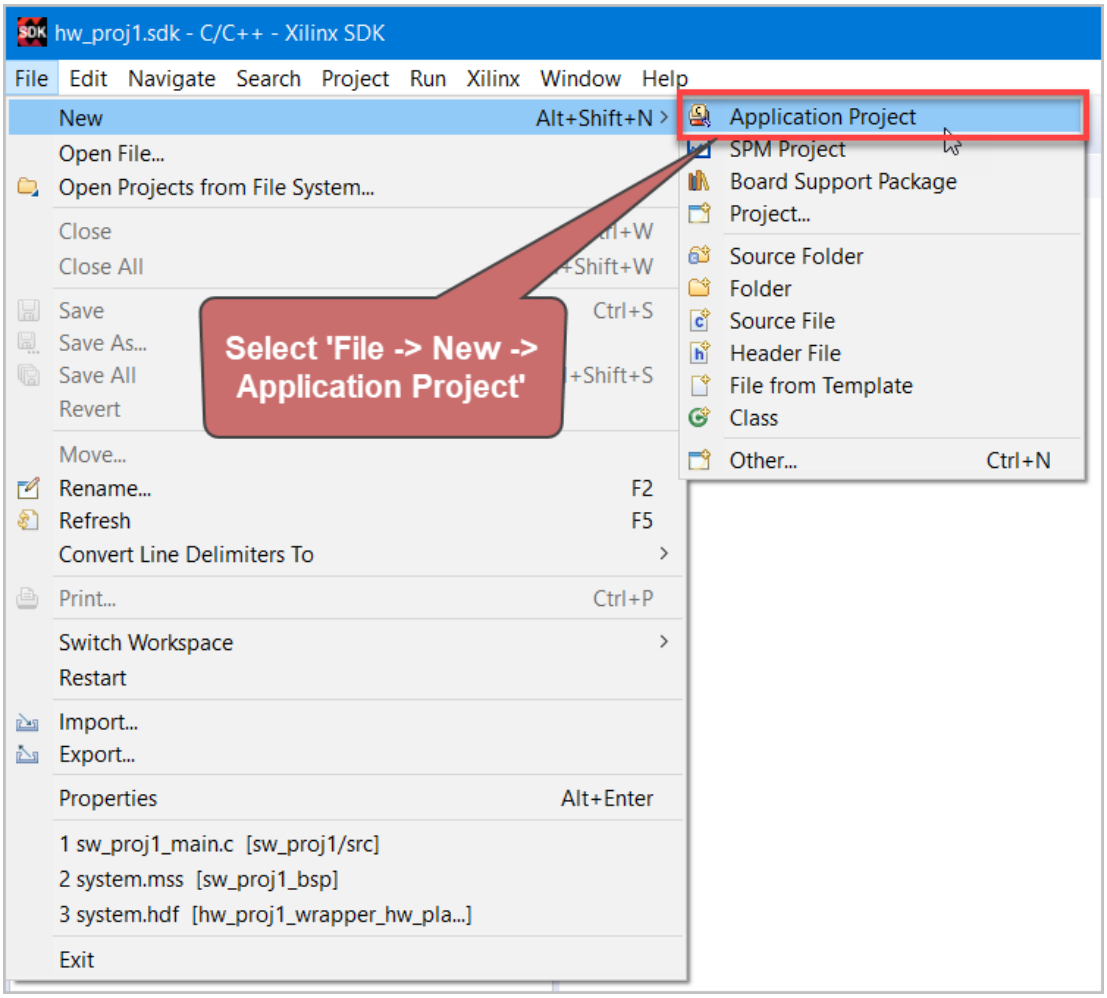


Figure 179. Select File->New-> Application Project

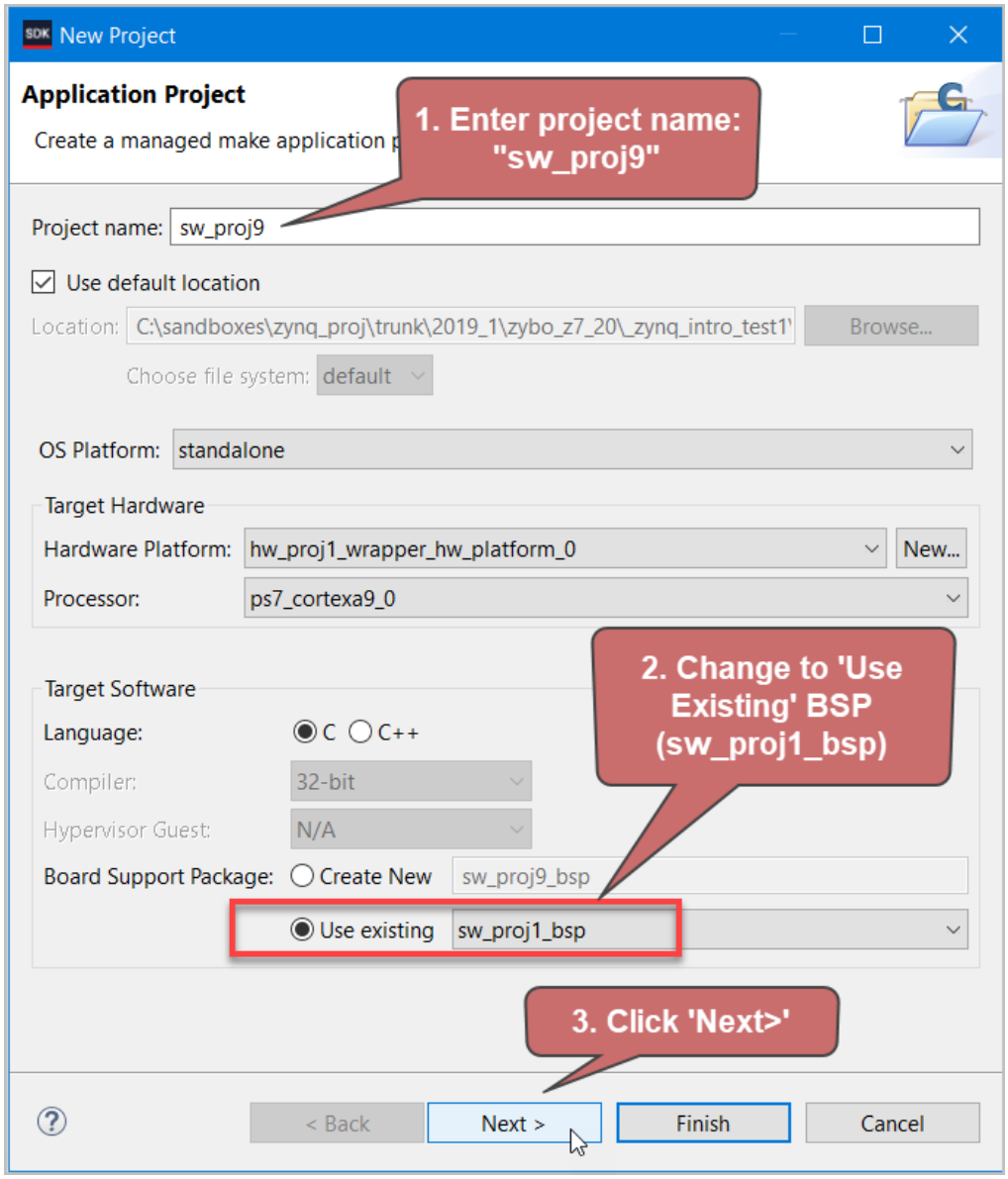


Figure 180. Enter the project details (part 1)

1. Enter the project name: sw_proj9
2. Board Support Package: Use existing
3. Click Next (don't click Finish!)

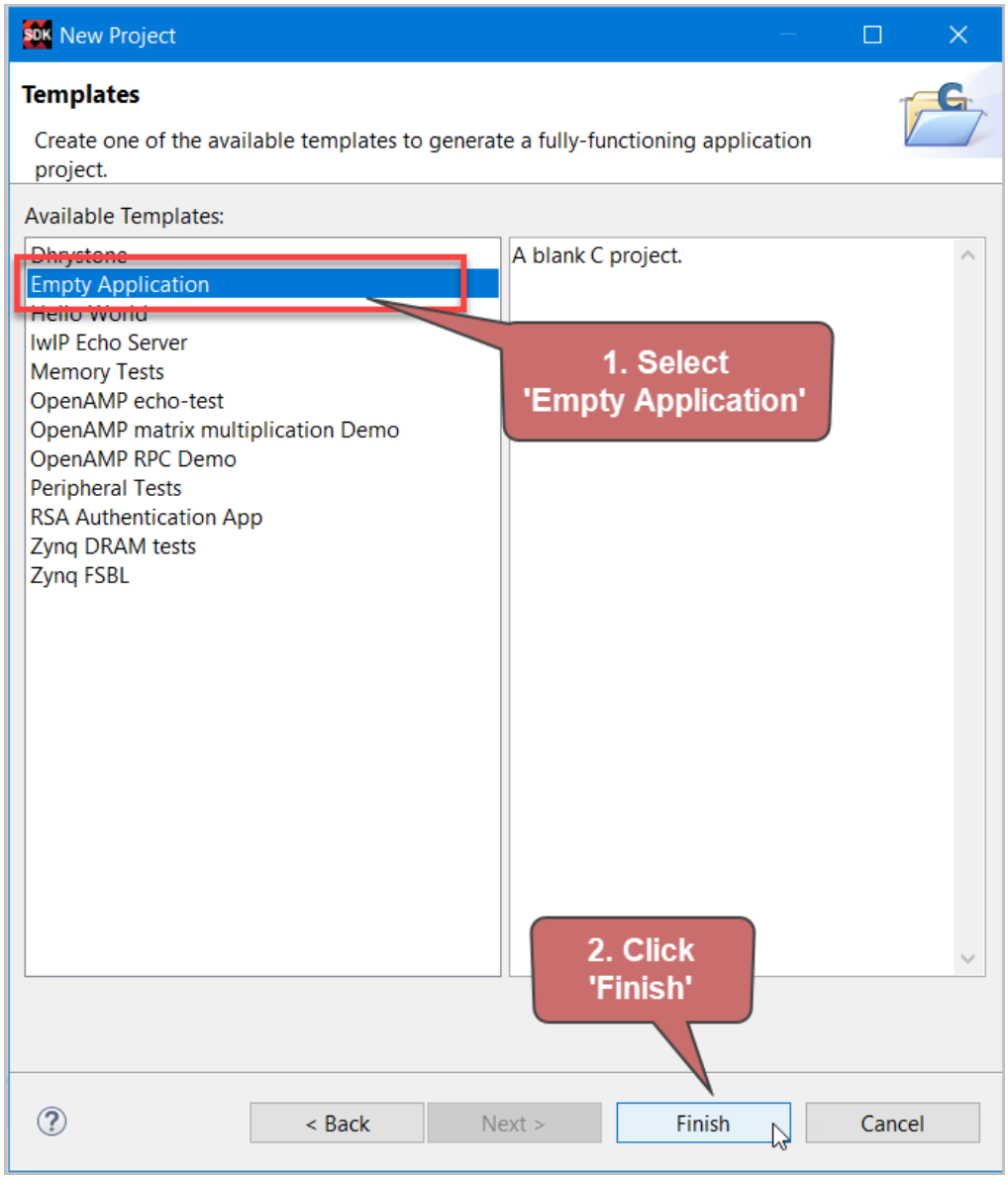


Figure 181. Enter the project details (part 2)

1. Select Empty Application
2. Click Finish

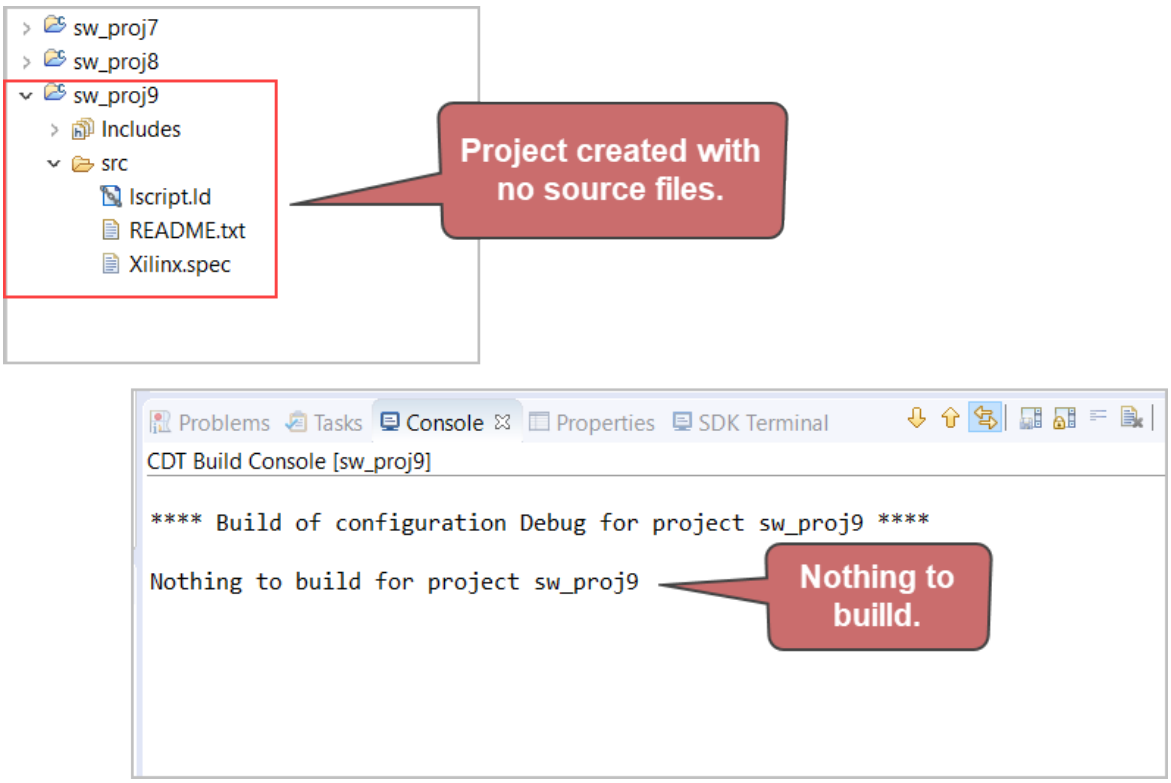


Figure 183. Project created with no source files

A project named “sw_proj9” is created with no project files. The CDT Build Console confirms that there is nothing to build.

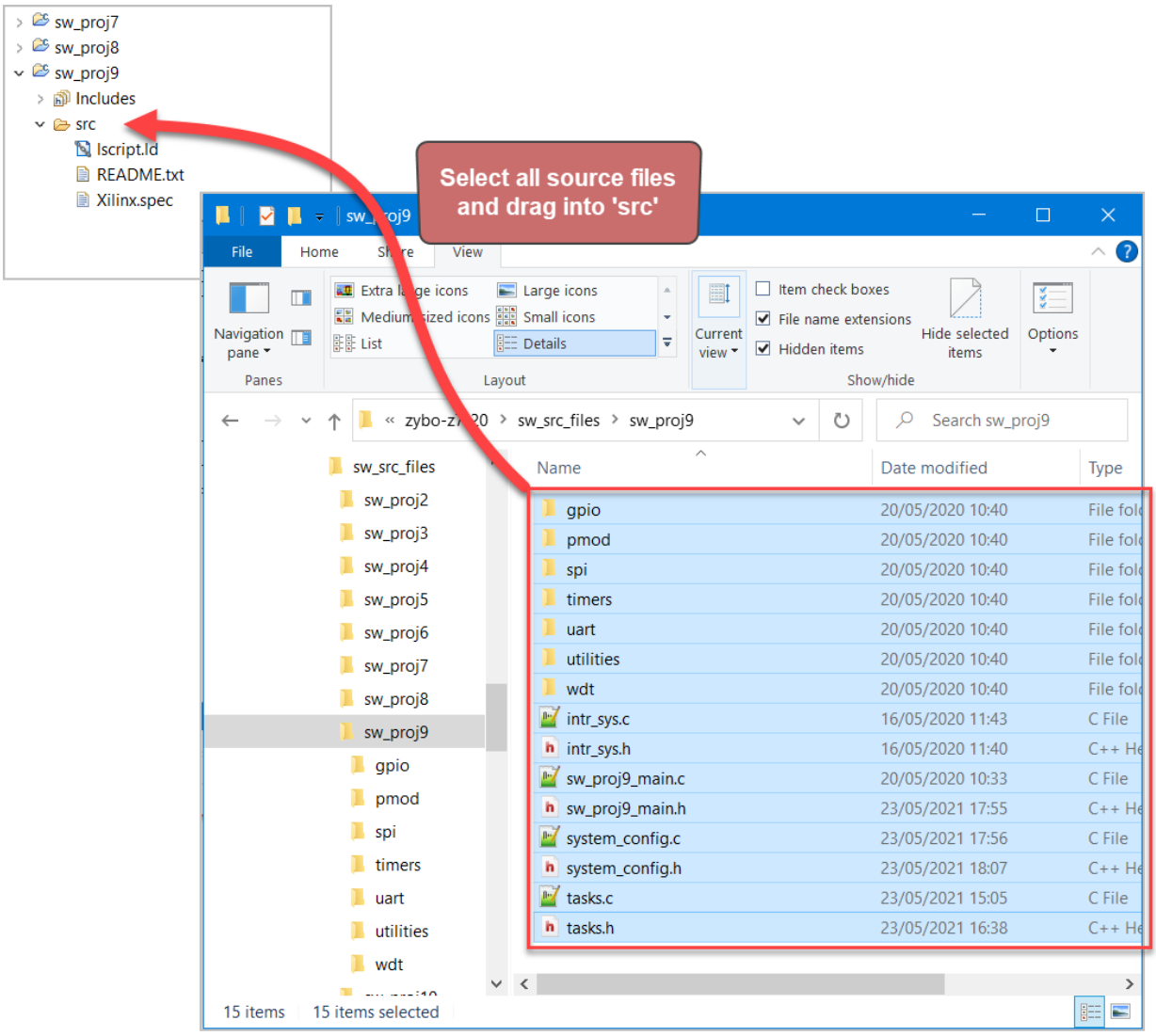


Figure 184. Drag the source files from Windows Explorer into the SDK project

Open the location where the source files for software project 9 are saved on your PC. Drag all the files and directory and from Windows explorer directly into the "**src**" folder in SDK.

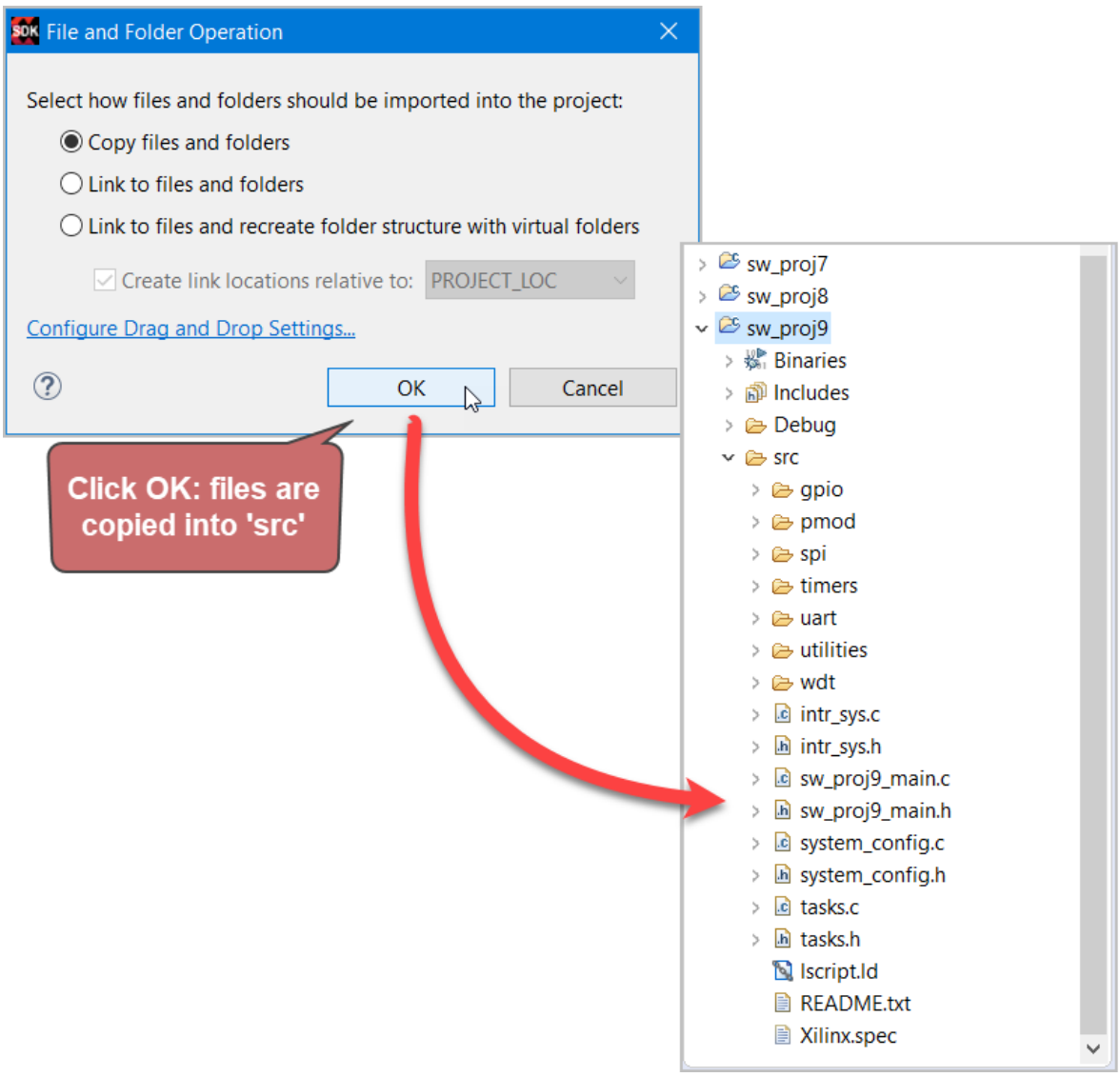


Figure 185. Click OK, and the files will be copied into the project.

In the File Operation dialog box, ensure “Copy files and folders” is checked, and click OK.

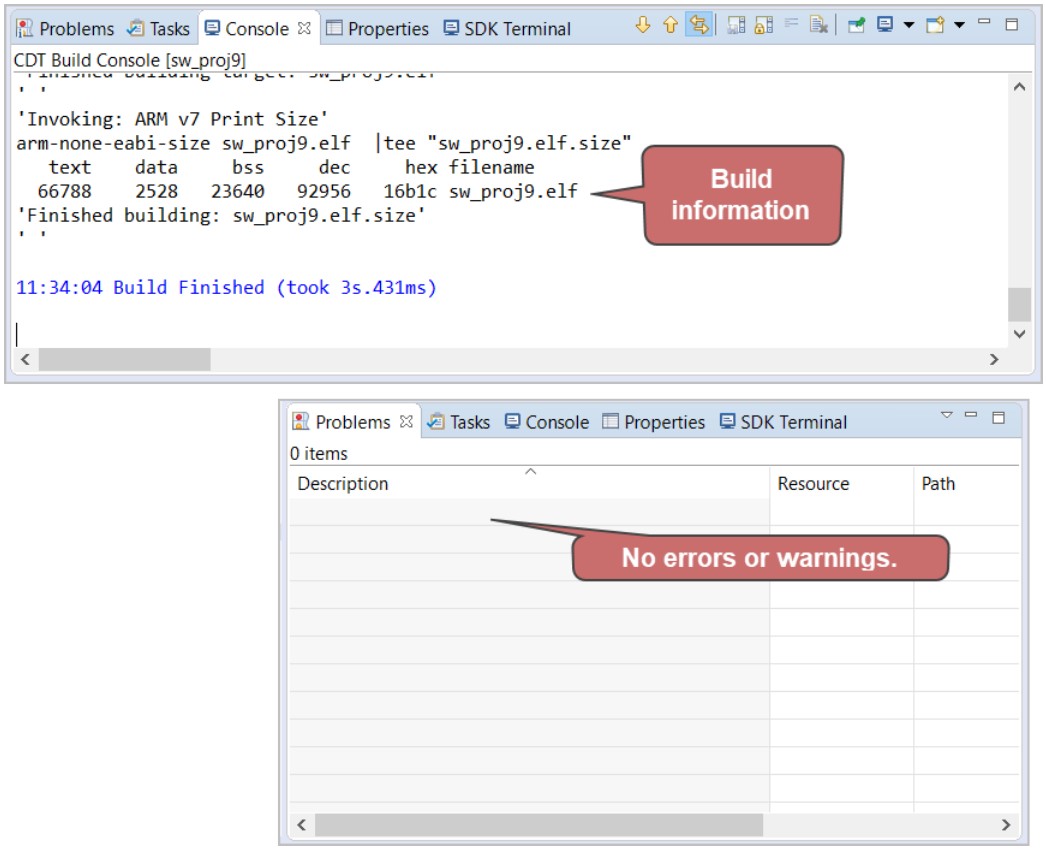


Figure 186. The project should build successfully

If "Build Automatically" is selected, the project should build successfully, with no errors or warnings.

[Exceptions: SDK 2018.3/SDK 2019.1 may indicate a warning as follows, which can be ignored:

#pragma message: For the sleep routines, Global timer is being used

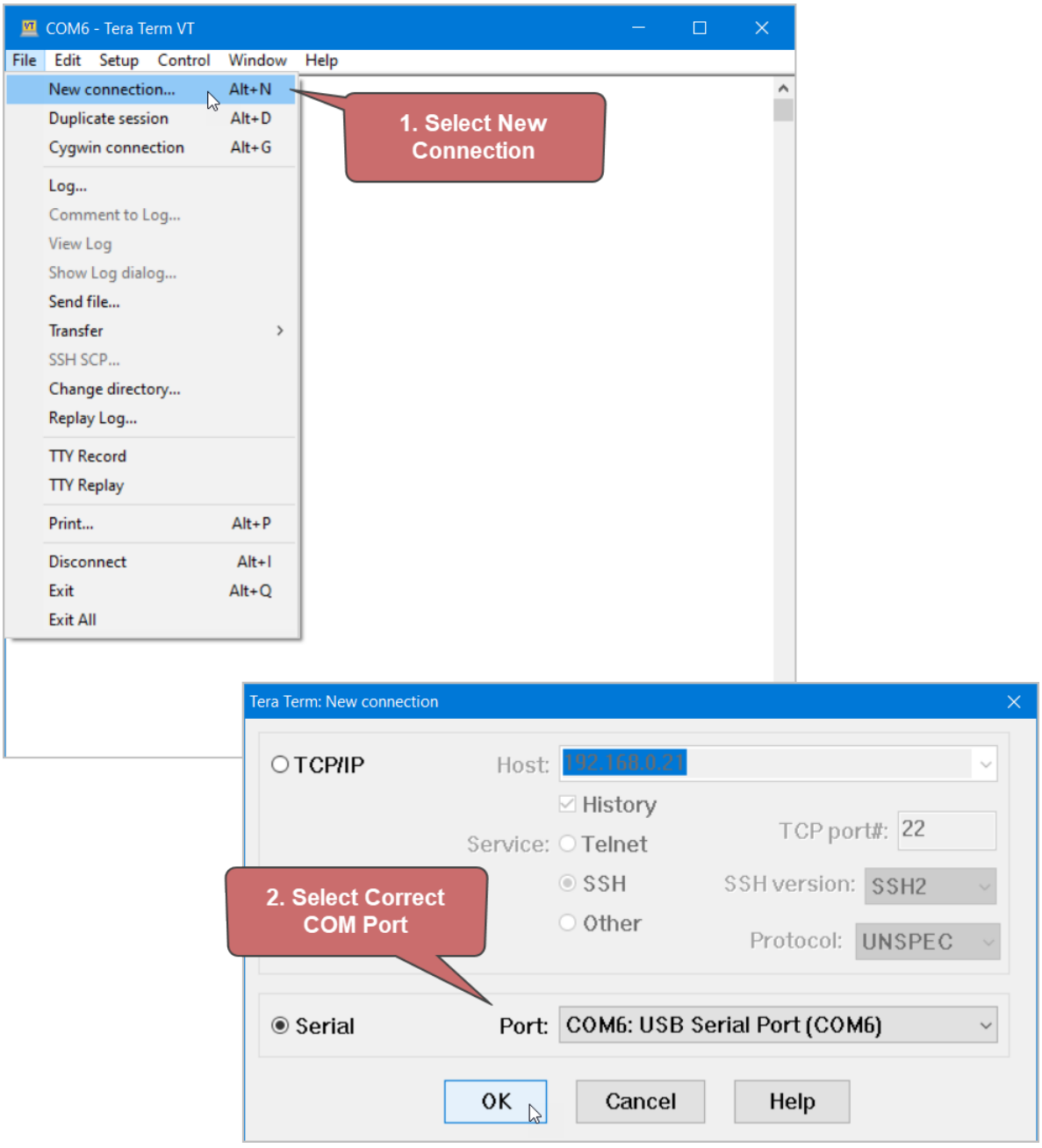


Figure 187. Reconnect terminal to platform if disconnected in last project.

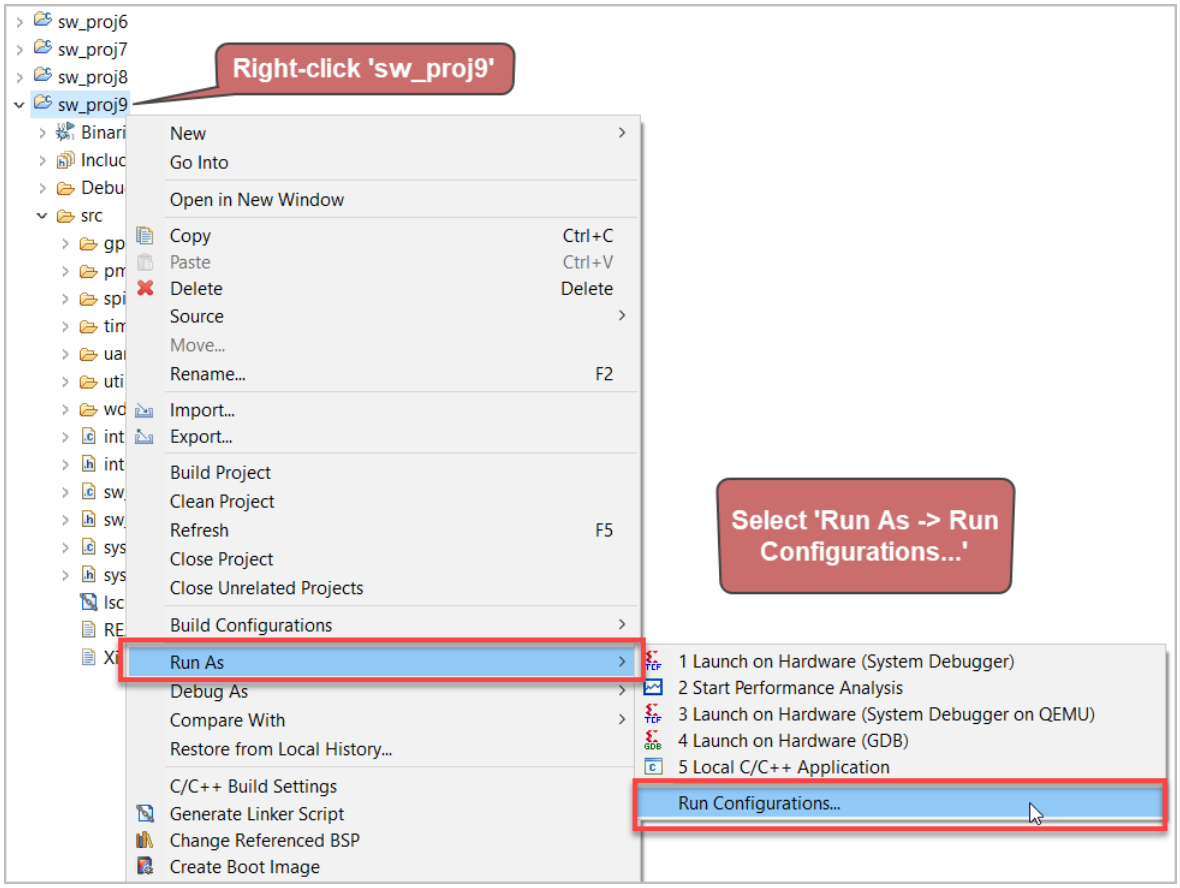


Figure 188. Right-click the software project and create a Run Configuration

To run the program, the first step is to create a Run configuration. Right-click on the project, and select the 'Run Configurations...' option.

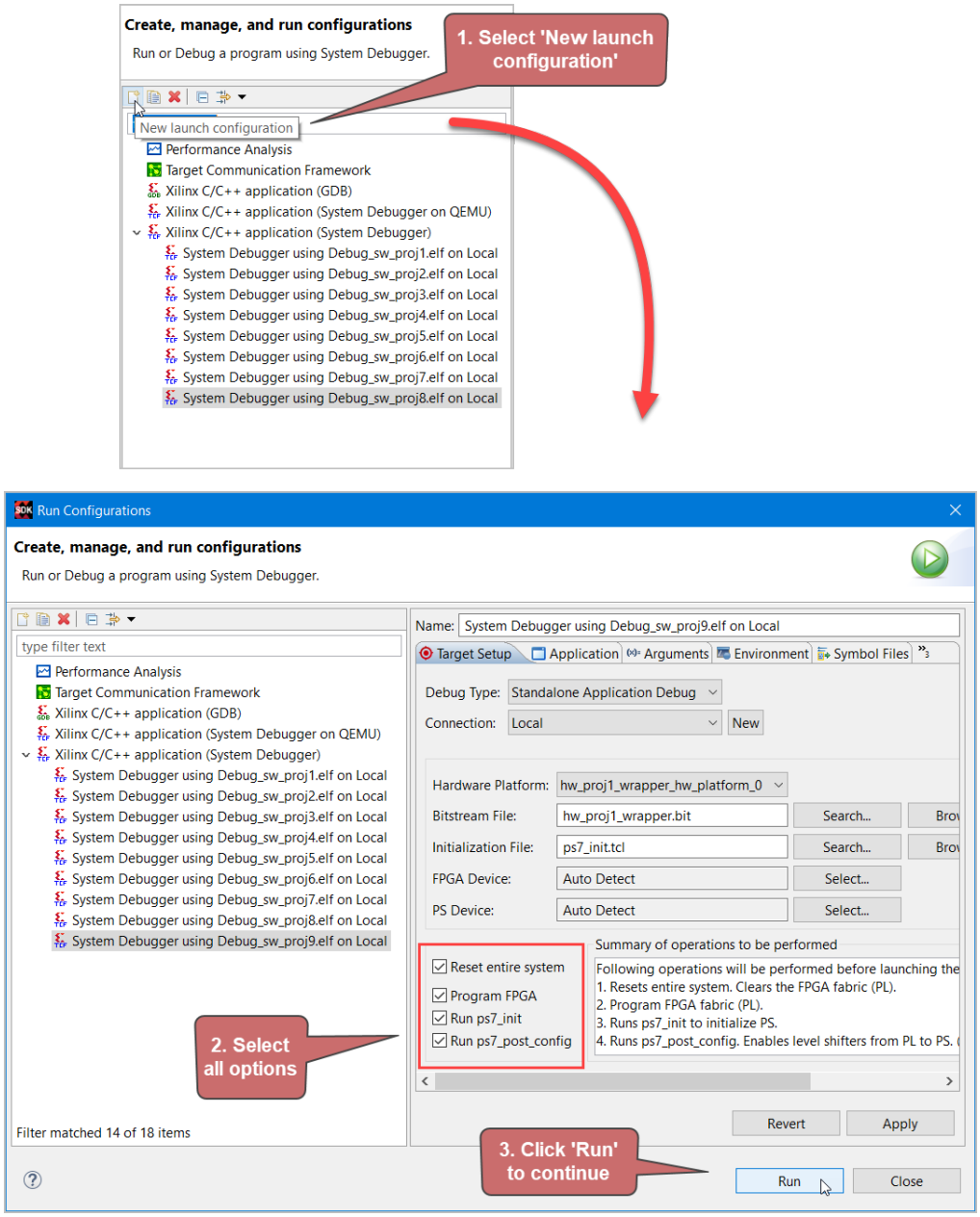


Figure 189. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

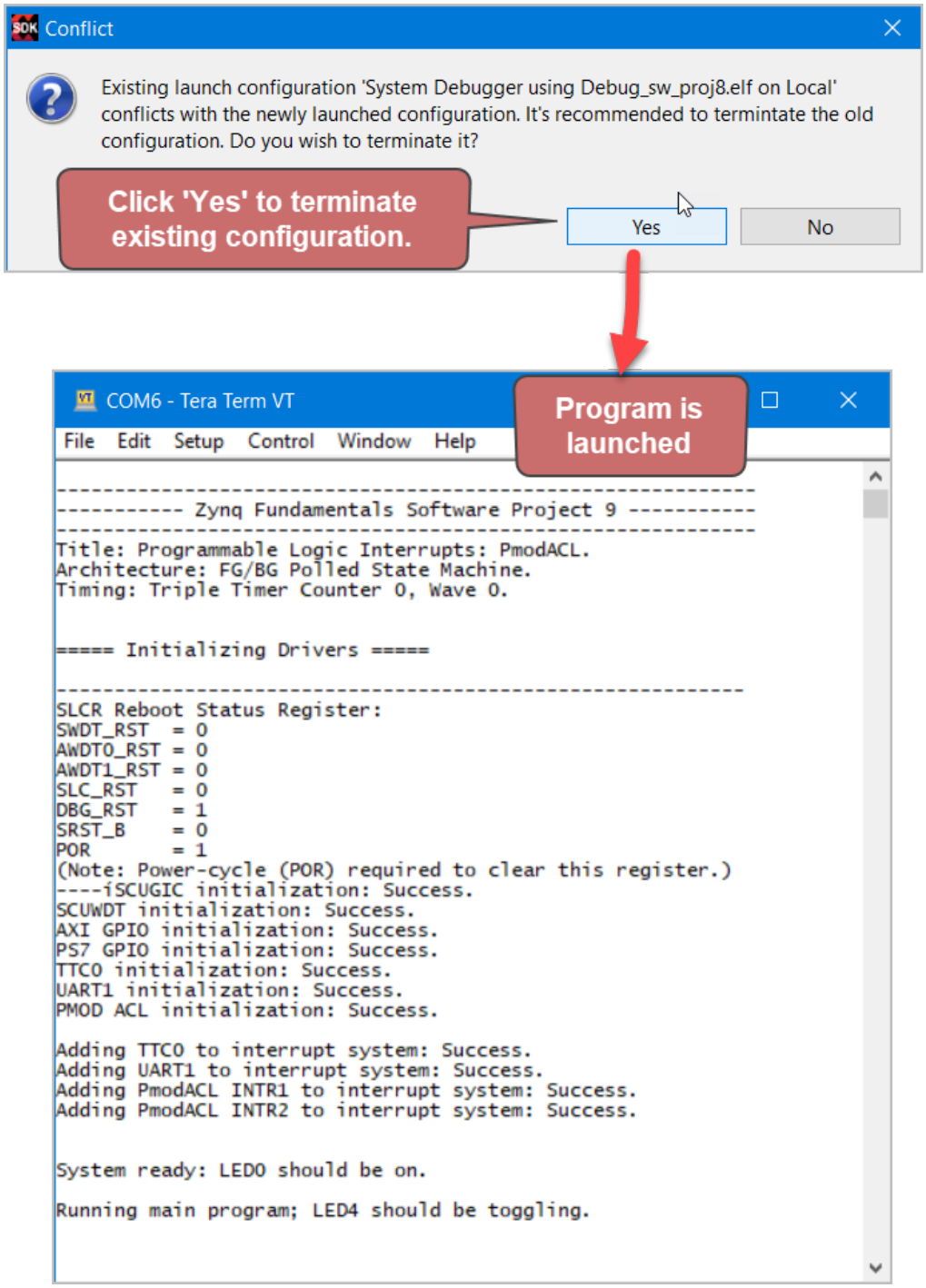


Figure 190. Terminal output for Software Project 9

If prompted, select 'Yes' to terminate any existing configuration. The FPGA will be programmed and the application will be downloaded to the processor. The terminal should display the results for launching Software Project 9.

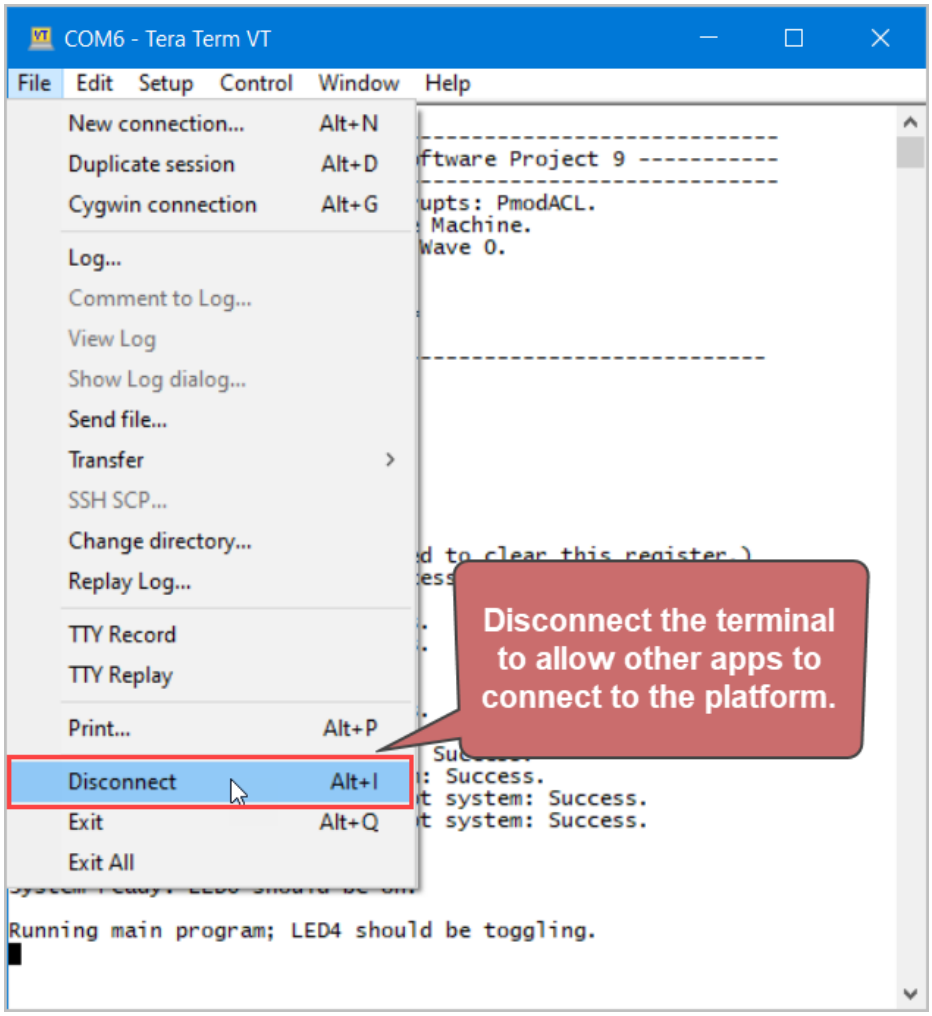


Figure 191. Disconnect the platform.

If the user wants to communicate with the command handler on the platform using a host application, the terminal needs to be disconnected.

3.10 Software Project 10: Additional Interrupt Topics

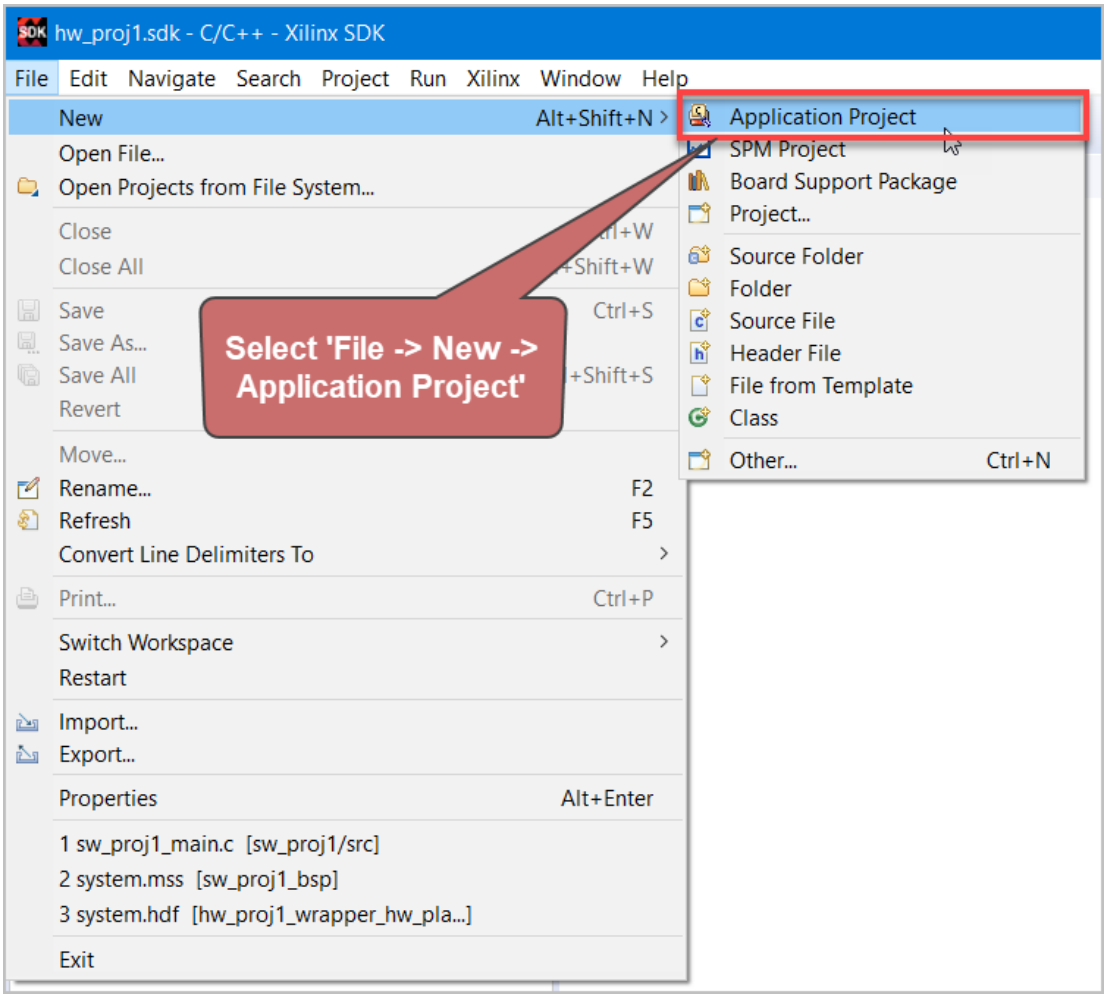


Figure 192. Select File->New-> Application Project

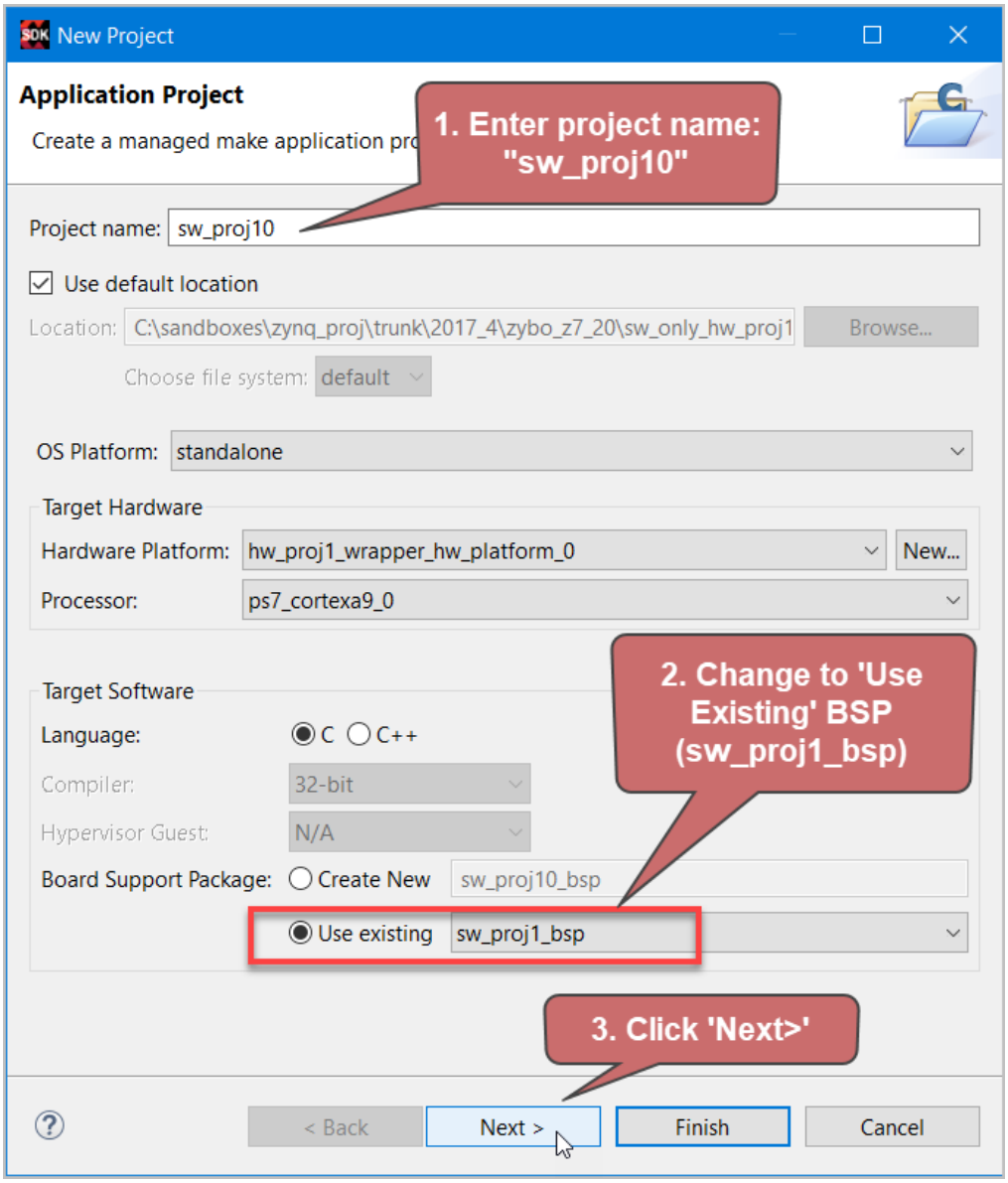


Figure 193. Enter the project details (part 1)

1. Enter the project name: sw_proj10
2. Board Support Package: Use existing
3. Click Next (don't click Finish!)

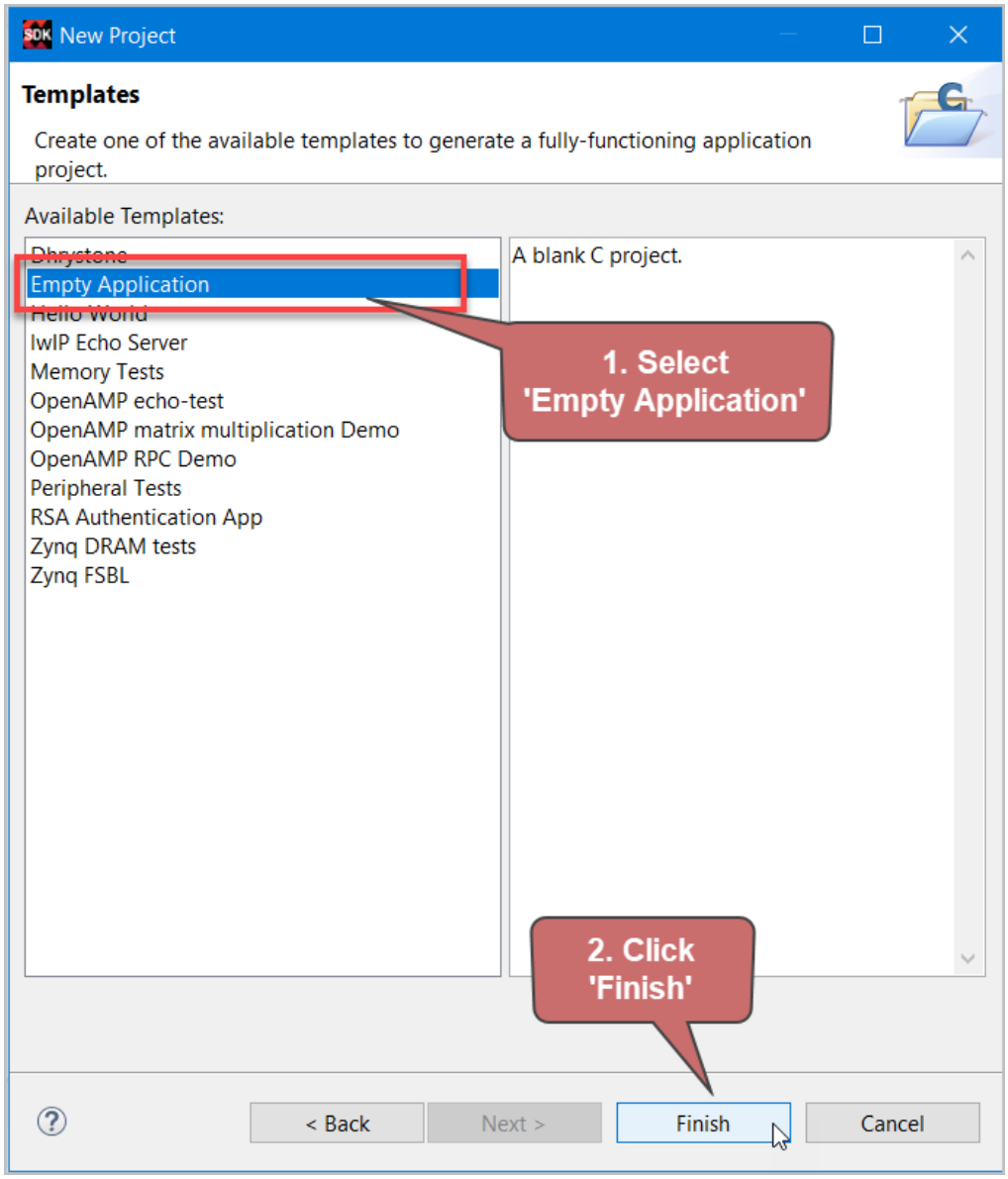


Figure 194. Enter the project details (part 2)

1. Select Empty Application
2. Click Finish

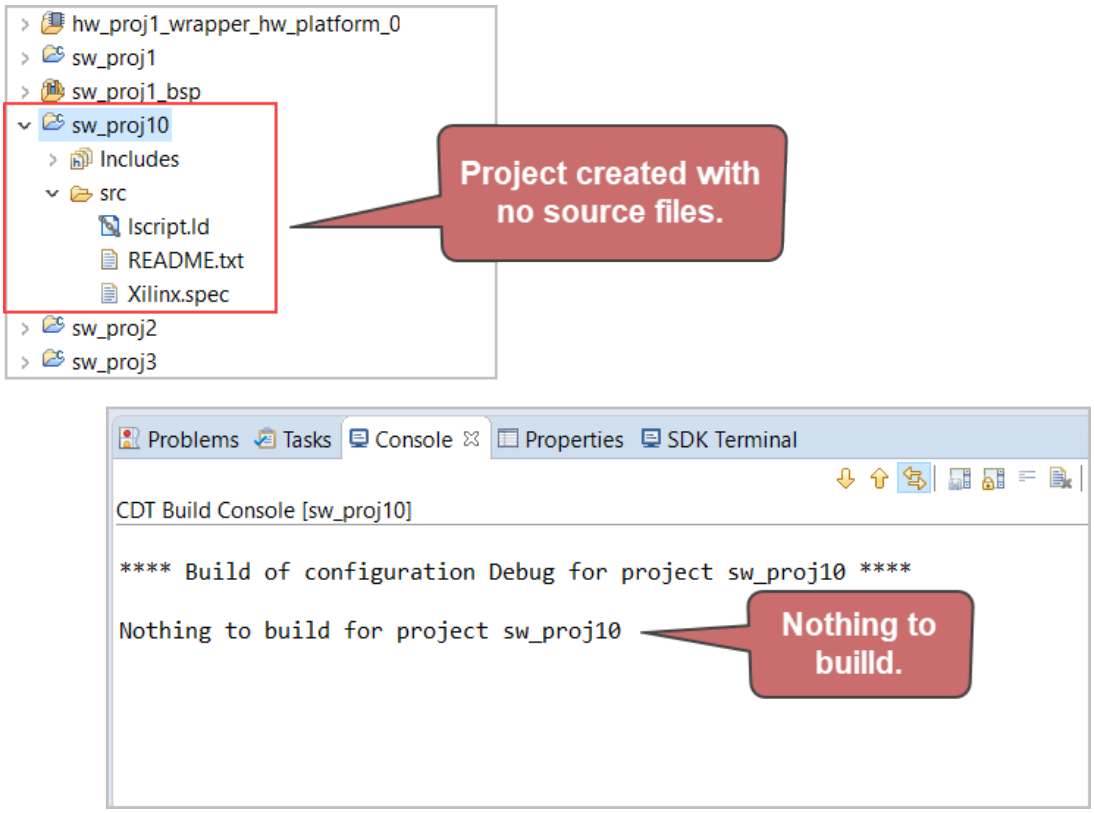


Figure 196. Project created with no source files

A project named "sw_proj10" is created with no project files. The CDT Build Console confirms that there is nothing to build.

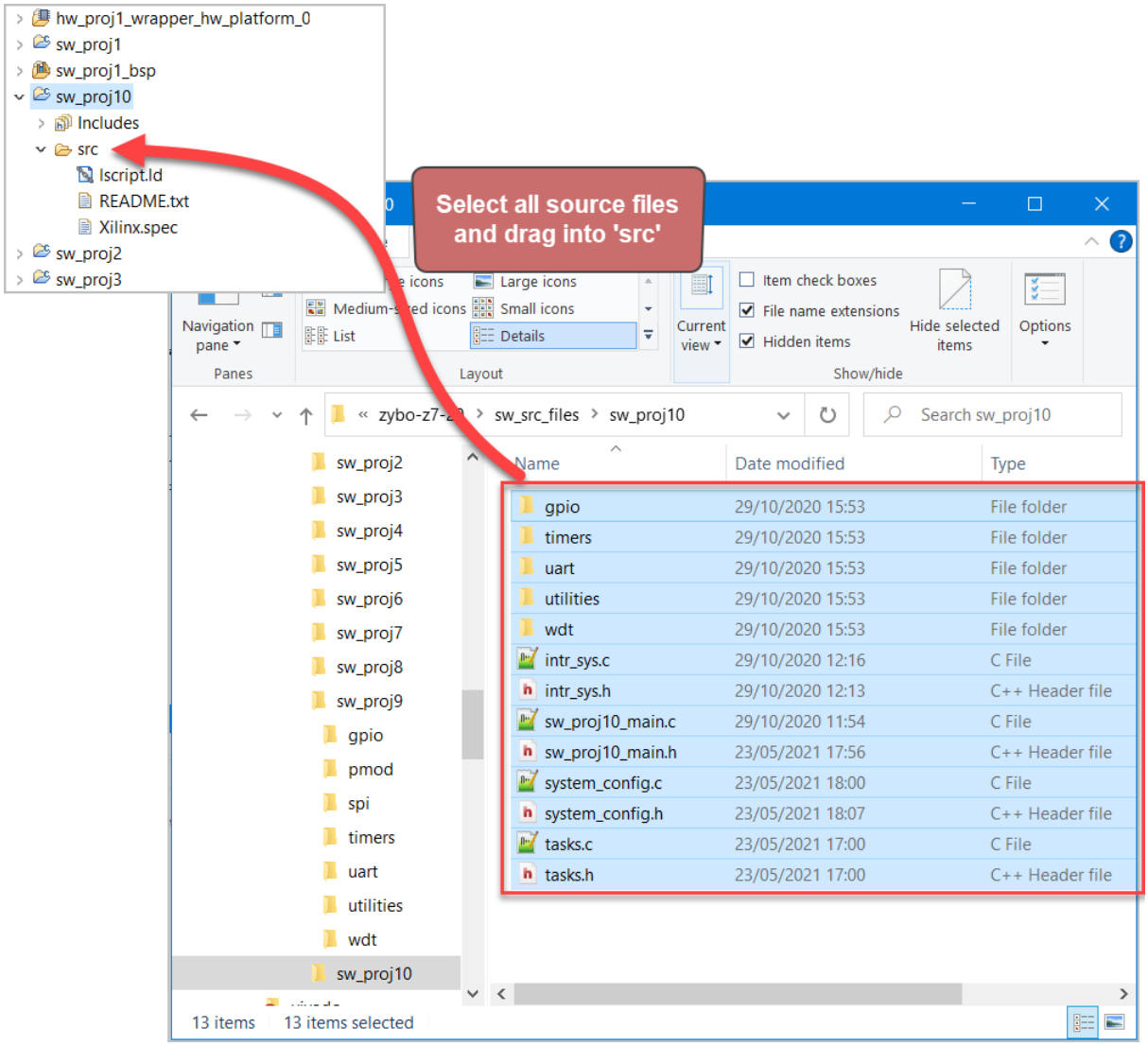


Figure 197. Drag the source files from Windows Explorer into the SDK project

Open the location where the source files for software project 10 are saved on your PC. Drag all the files and directory and from Windows explorer directly into the "**src**" folder in SDK.

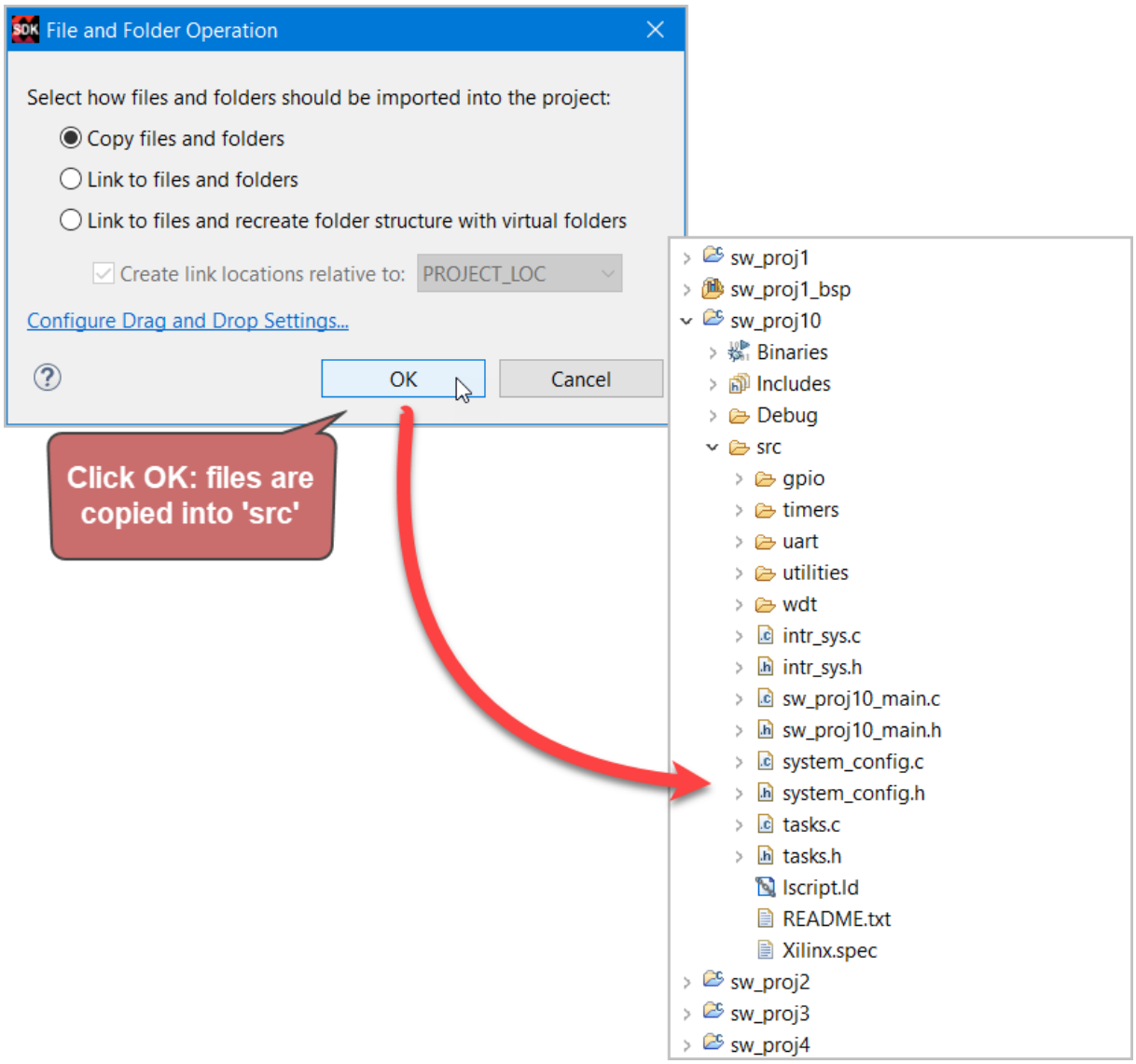


Figure 198. Click OK, and the files will be copied into the project.

In the File Operation dialog box, ensure “Copy files and folders” is checked, and click OK.

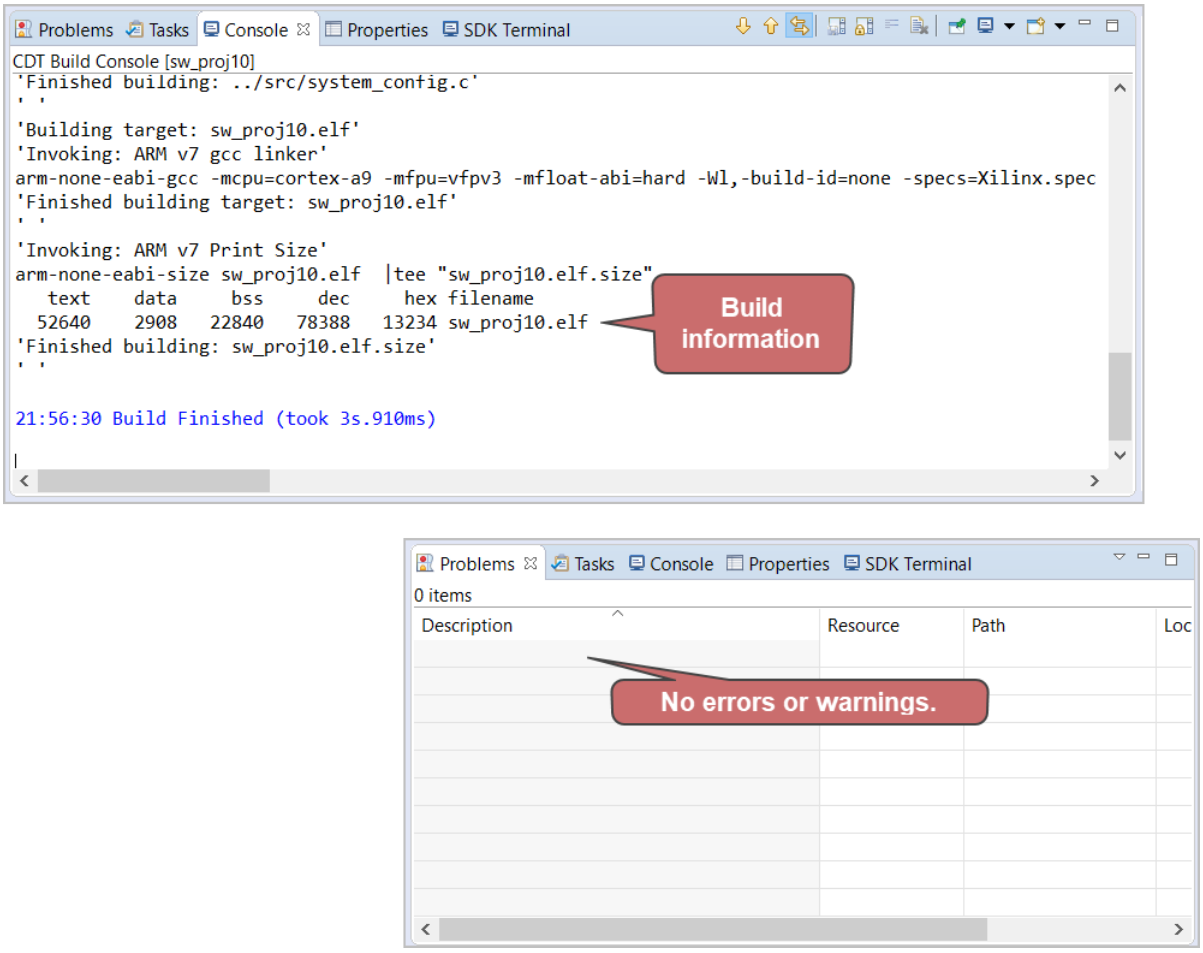


Figure 199. The project should build successfully

If "Build Automatically" is selected, the project should build successfully, with no errors or warnings.

[Exceptions: SDK 2018.3/SDK 2019.1 may indicate a warning as follows, which can be ignored:

#pragma message: For the sleep routines, Global timer is being used

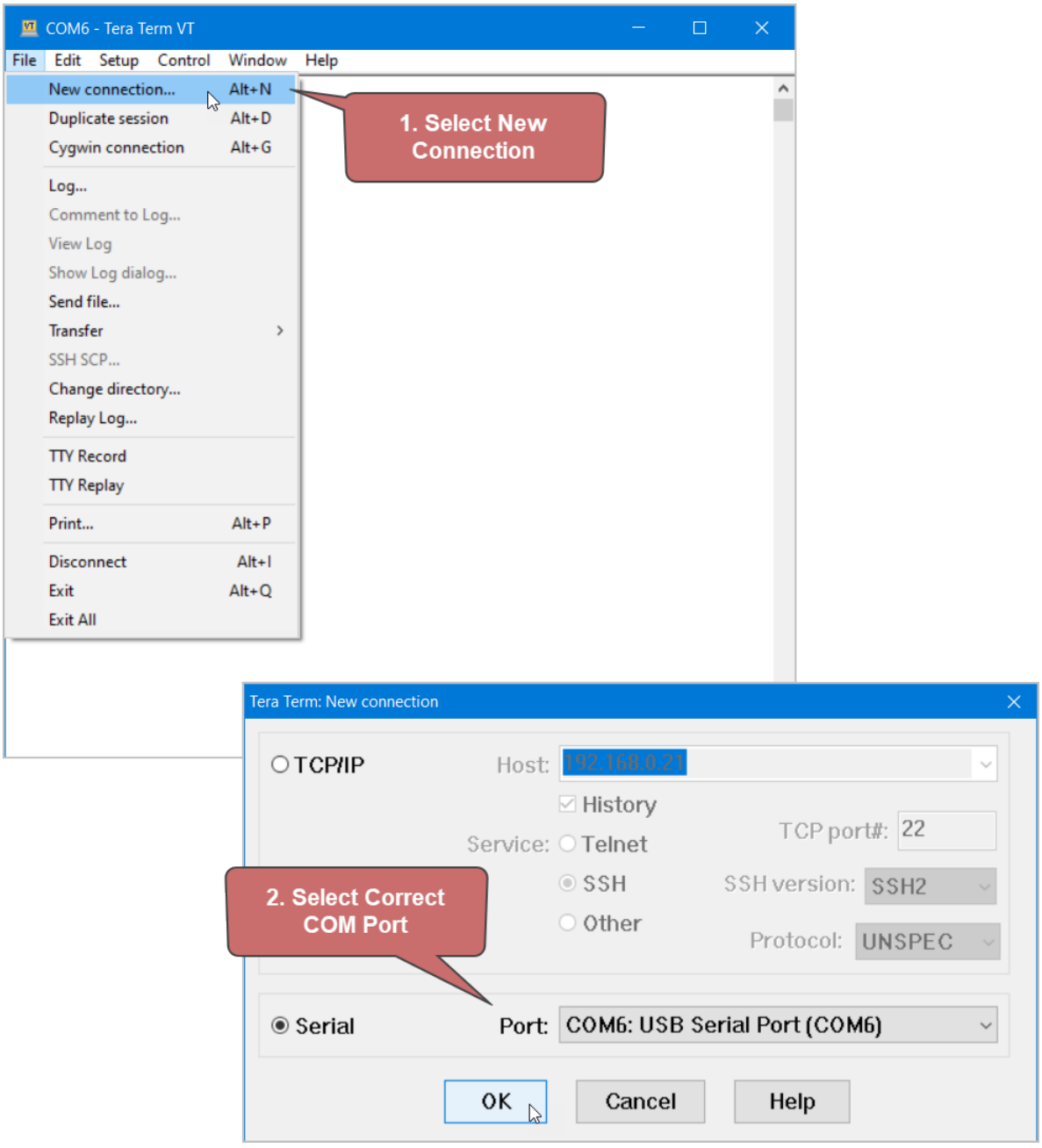


Figure 200. Reconnect terminal to platform if disconnected in last project.

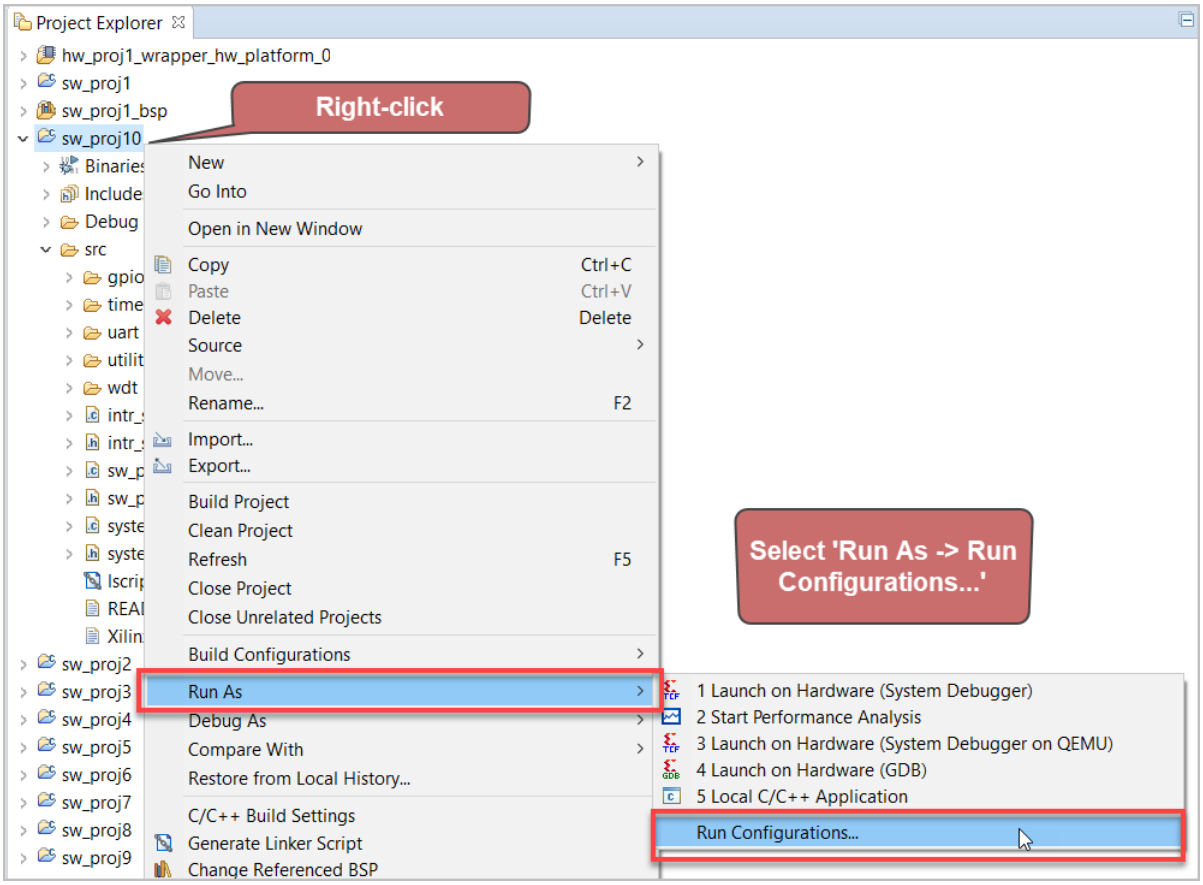


Figure 201. Right-click the software project and create a Run Configuration

To run the program, the first step is to create a Run configuration. Right-click on the project, and select the 'Run Configurations...' option.

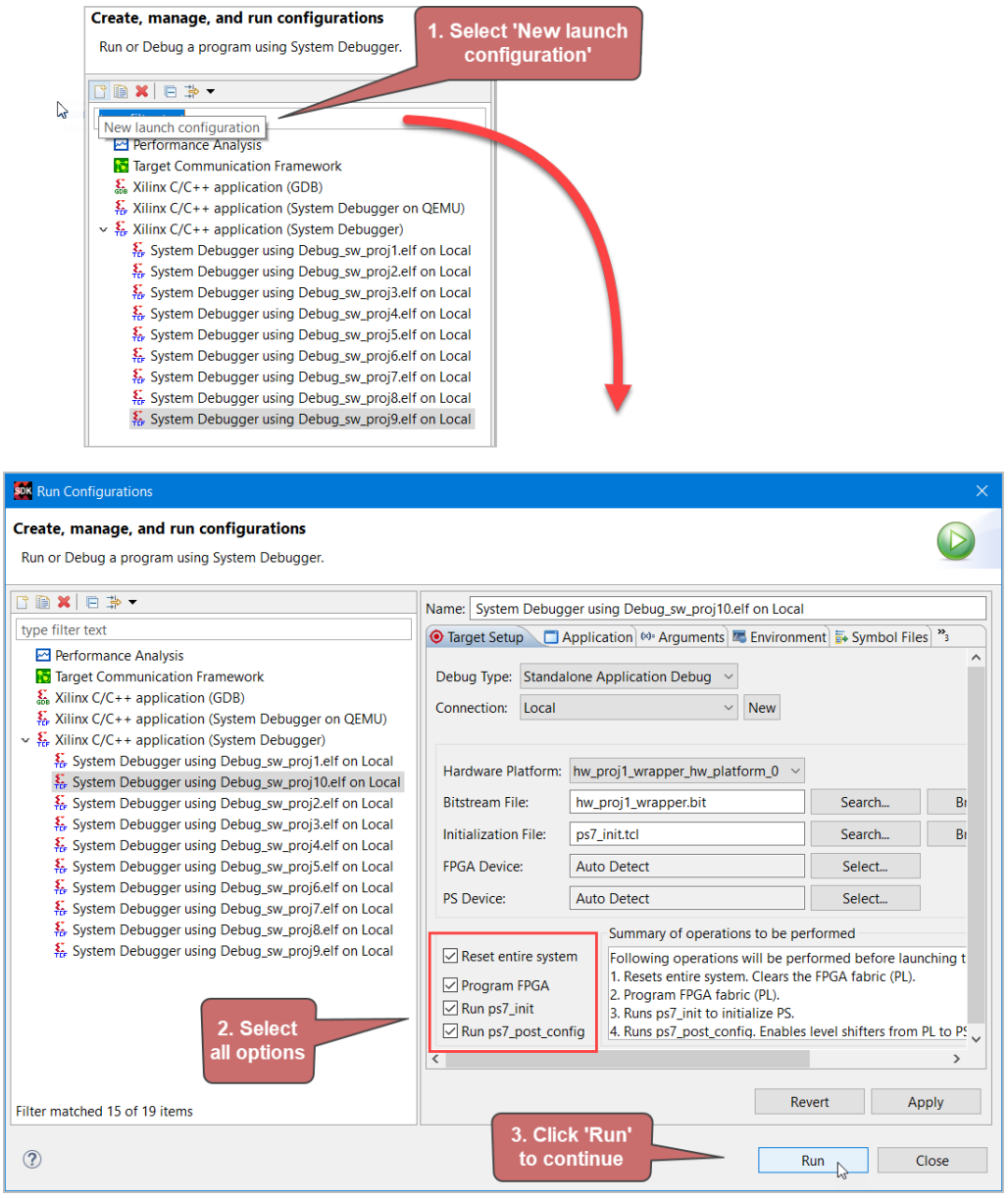


Figure 202. Create a TCF (i.e. System Debugger) option and click Run to execute the program.

1. Ensure the Xilinx C/C++ application (System Debugger) is selected.
2. Click on New Launch Configuration.
3. Select all options:
    a. Reset entire system
    b. Program FPGA
    c. Run ps7_init
    d. Run ps7_post_config

4. Click on 'Run' to continue.

If the terminal was disconnected in the last project (to run the LabVIEW/Python applications) then the Tera Term connection should be re-opened.

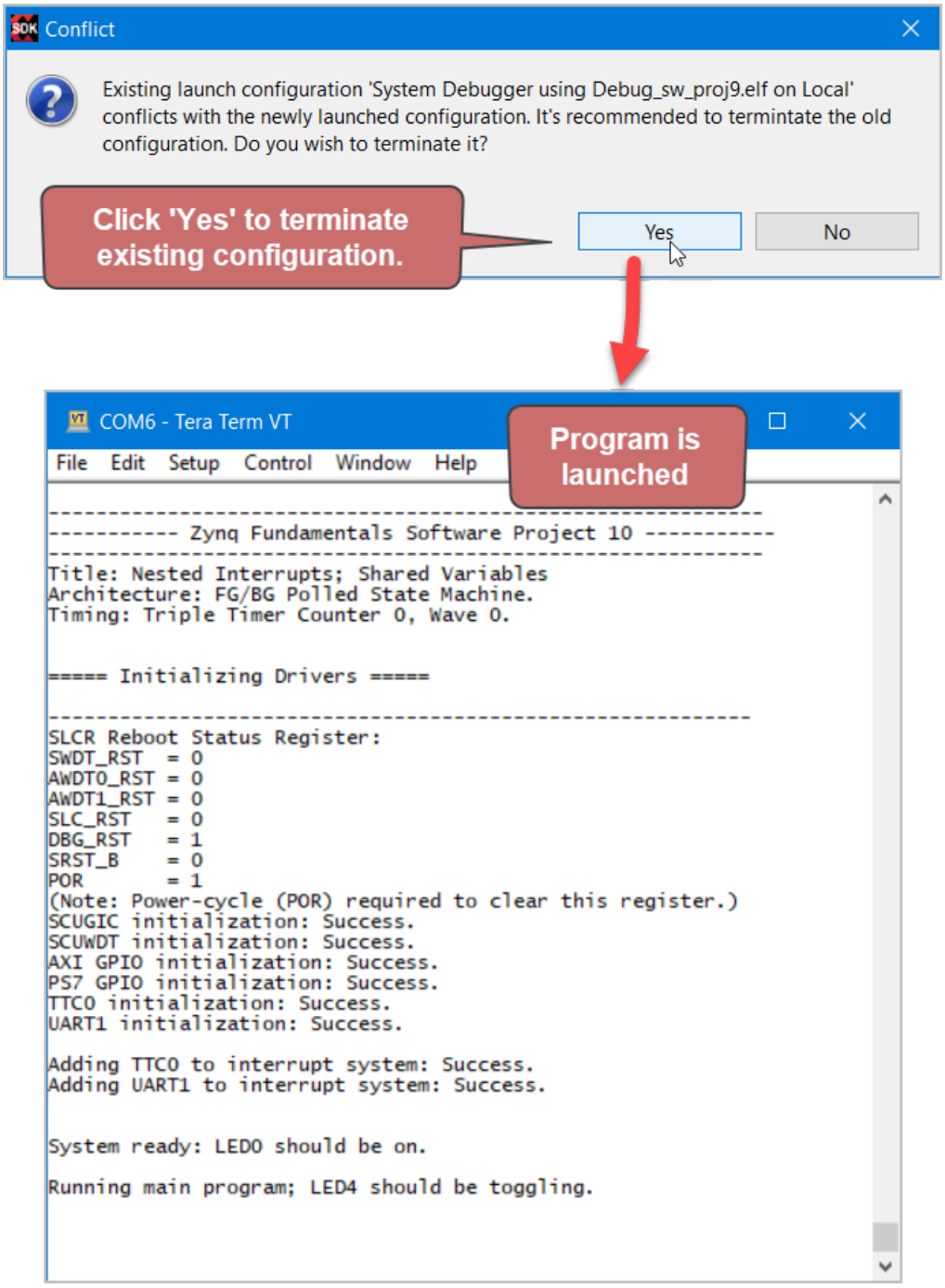


Figure 203. Terminal output for Software Project 10

If prompted, select 'Yes' to terminate any existing configuration. The FPGA will be programmed and the application will be downloaded to the processor. The terminal program should display the results for launching Software Project 10.

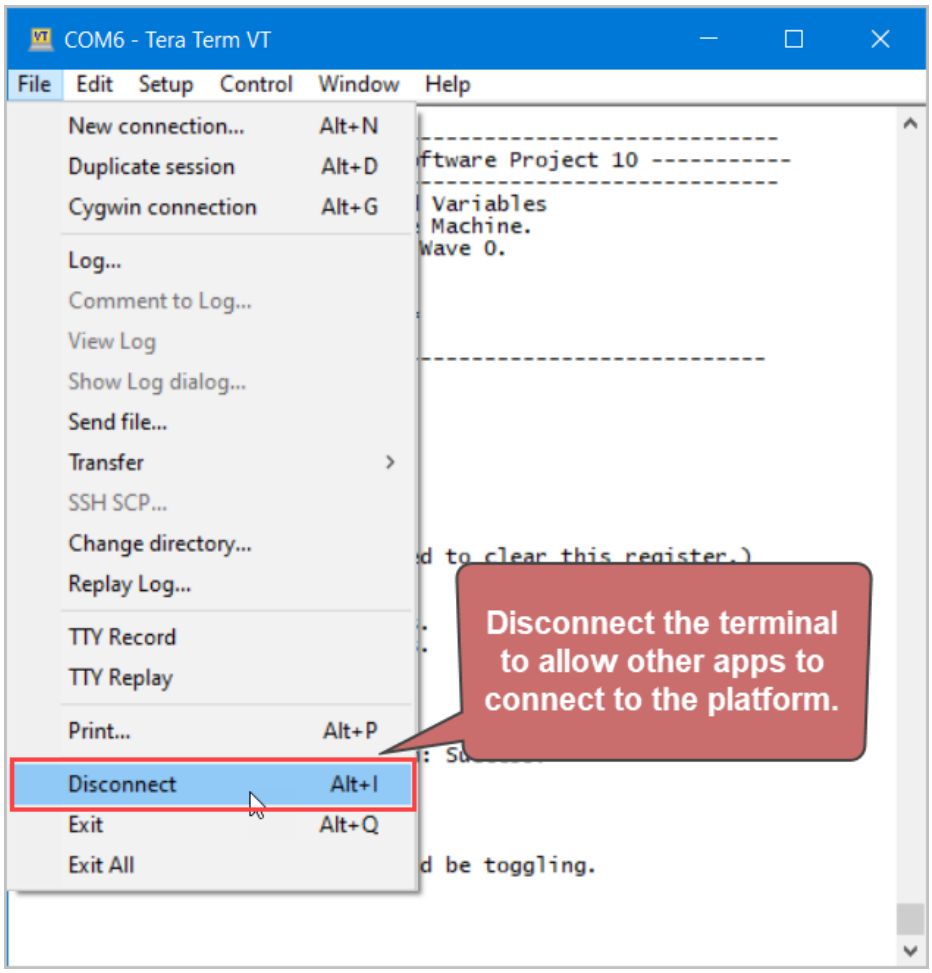


Figure 204. Disconnect the platform.

If the user wants to communicate with the command handler on the platform using a host application, the terminal needs to be disconnected.