Z-turn-Lite
入门指导
手册

版本 V1.0

2017 年 07 月 11 日

版本记录

版本号	说明	时间
V1.0	初始版本	2017/07/11

目录

1 概述

本教程将引导您使用 Vivado 2017.1 为 Z-Turn-Lite 开发板生成一个硬件平台，并在开发板运行多个应用程序来测试接口。

2 目标

通过本教程，您将能够：
•使用 Vivado 2017.1 为 Z-Turn-Lite 开发套件创建一个硬件平台
•将硬件平台导出到 SDK 2017.1
•在 SDK 中创建一个简单的“Hello World”应用程序，并在 Z-Turn-Lite 上运行
•在 SDK 中创建几个裸机程序，以测试 Z-Turn-Lite 上的几个外部接口
•从 microSD 卡启动 z-turn_lite
•从 QSPI 闪存启动 z-turn_lite

3 环境要求

本教程的软件和硬件需求：

3.1 软件

本参考设计的软件要求：
•Xilinx Vivado 2017.1 SDK
•USB 串口驱动程序
•putty 终端程序

3.2 硬件

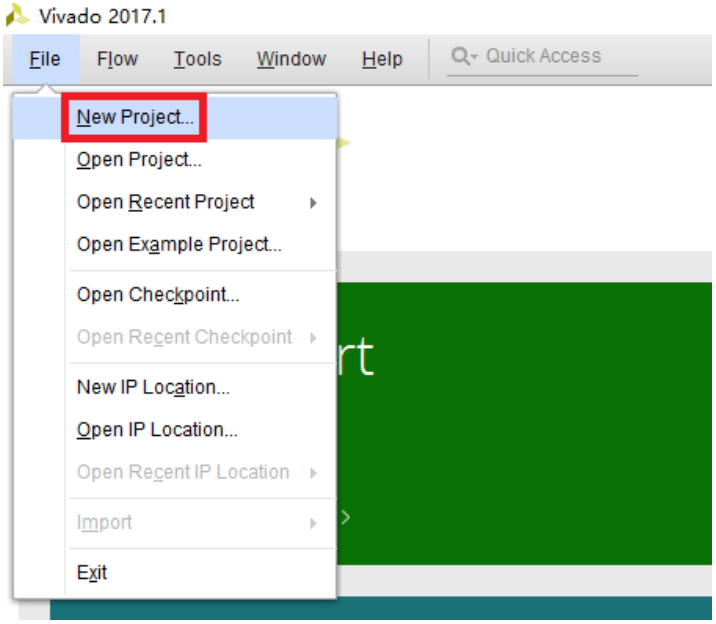
本参考设计的硬件要求：
•具有 4 GB RAM 和 1 GB 虚拟内存的计算机（推荐）
•Z-Turn-Lite 开发板
•5V 电源
•以太网电缆
•MY-UART012U USB 转 COM 线缆
•Xilinx USB Platform Cable DLC9 下载器

4 生成 z-turn-lite 开发板的硬件平台

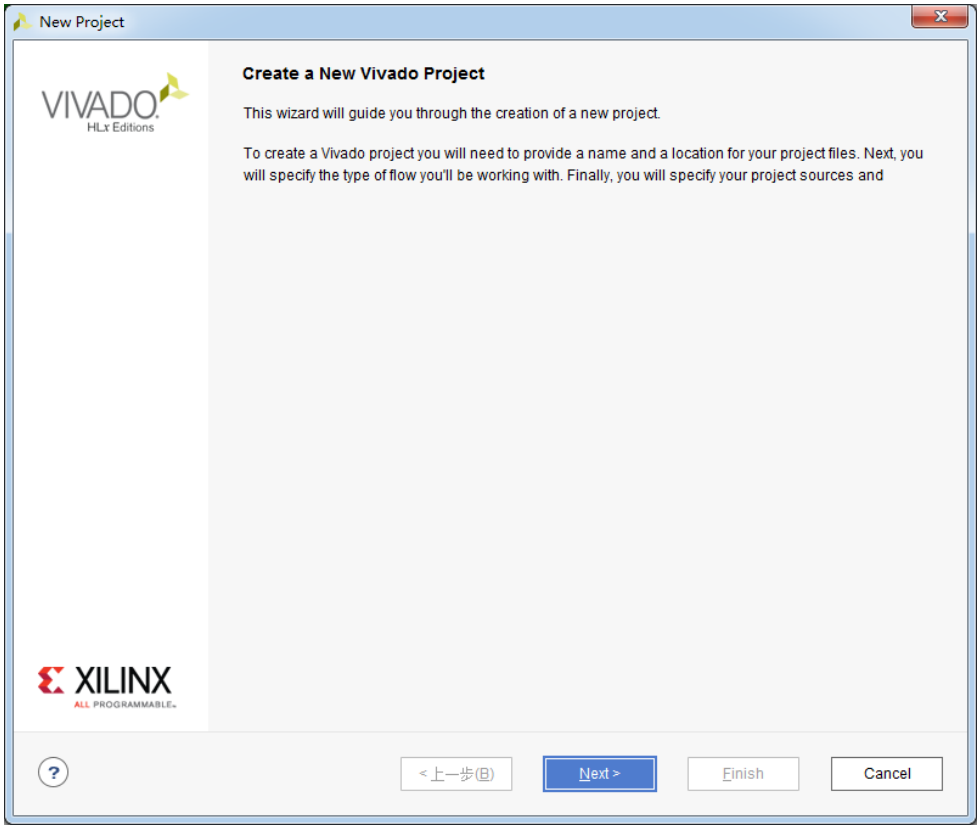
请按照以下步骤使用 Vivado 2017.1 工具的生成硬件平台。

- 单击电脑上的开始 **>** 所有程序 **> Xilinx Design Tools > Vivado 2017.1 >**Vivado 2017.1 启动 vivado

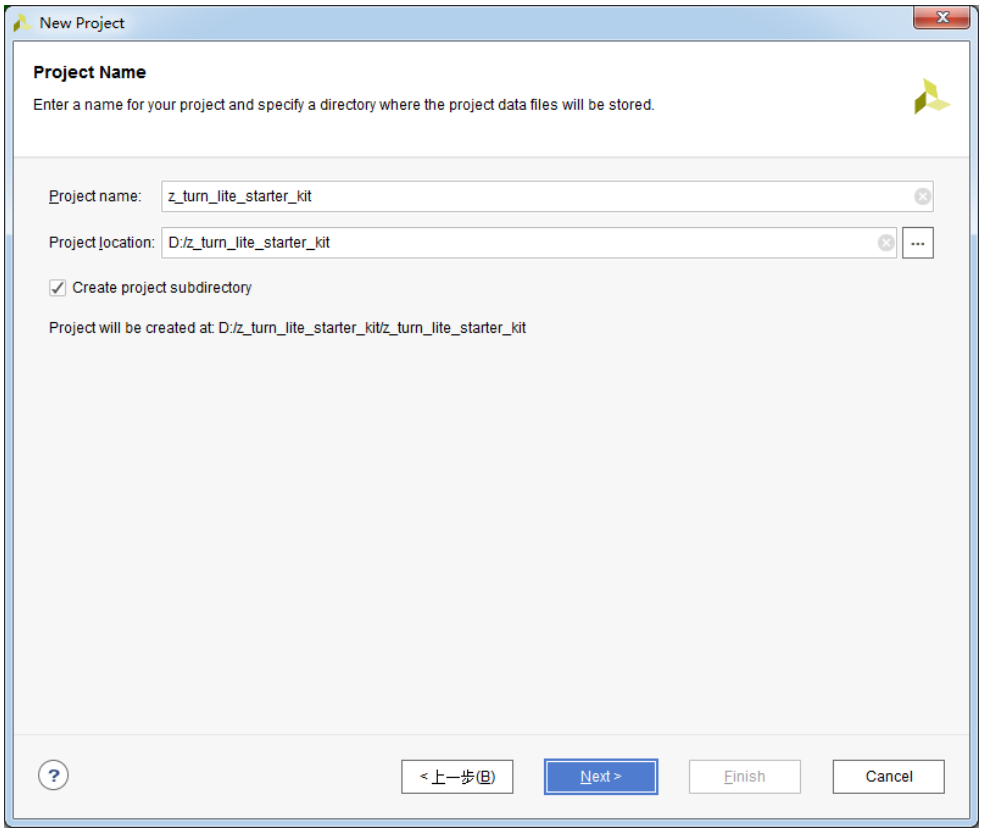
- 新建工程



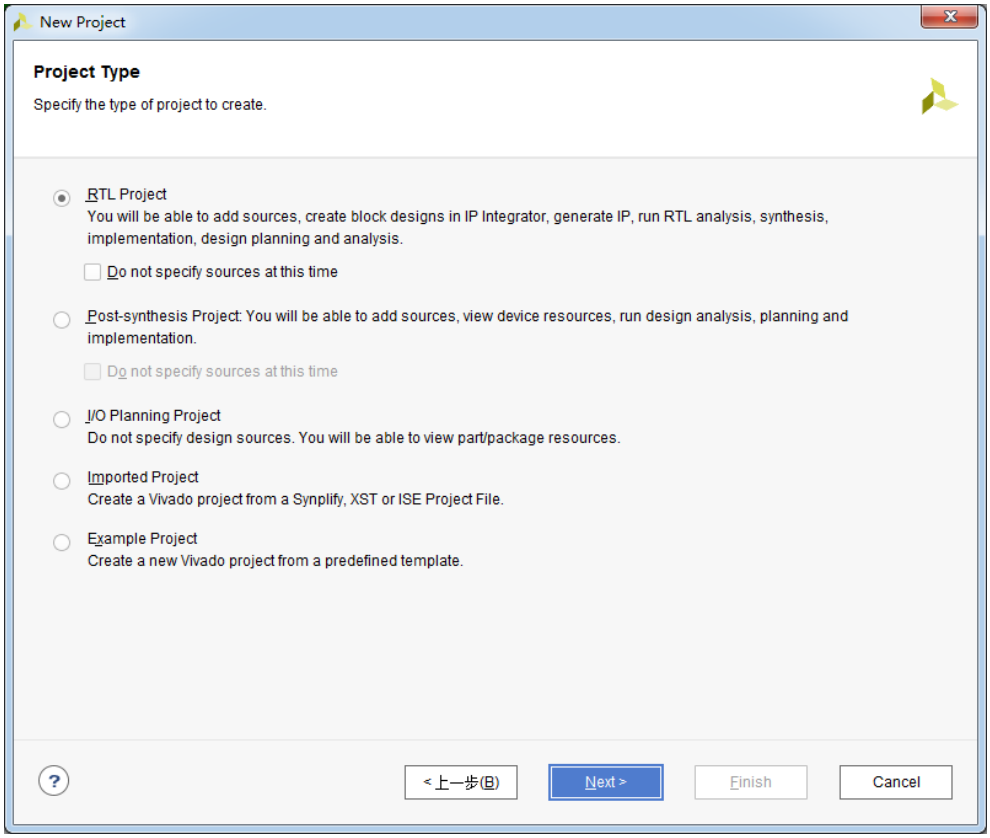
●单击 Next



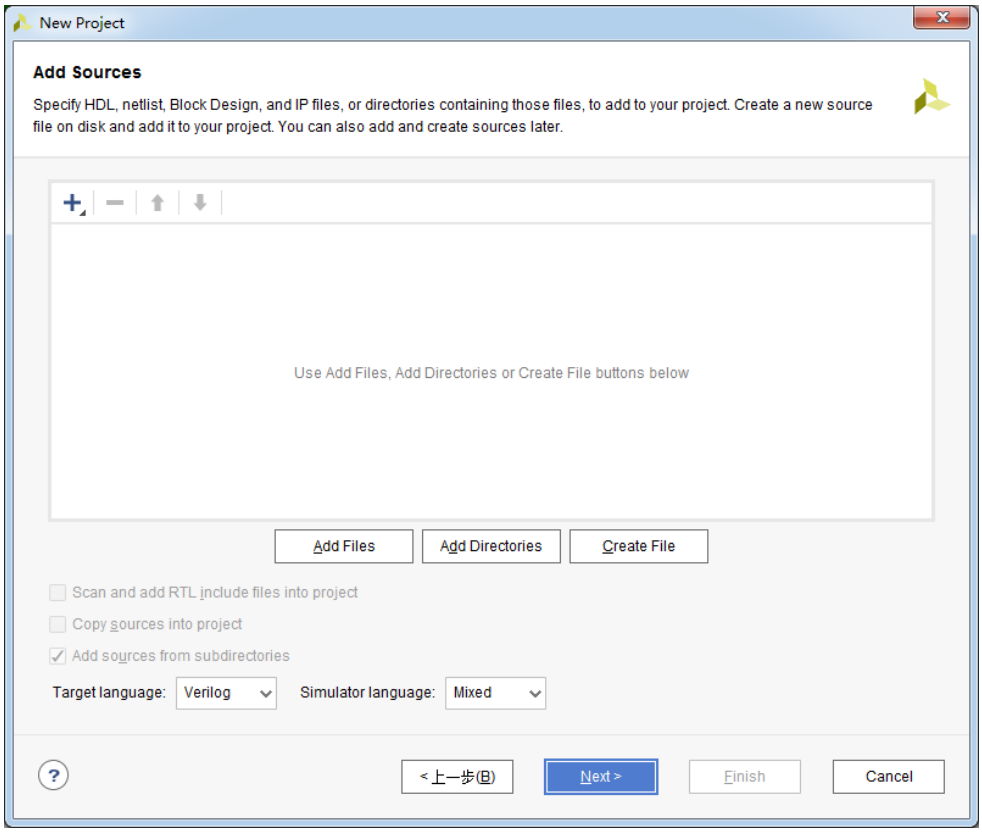
● 填写工程名称和工程的保存路径，单击 Next（注意路径中不要出现中文否则无法识别，而且保存路径也不要太长）。



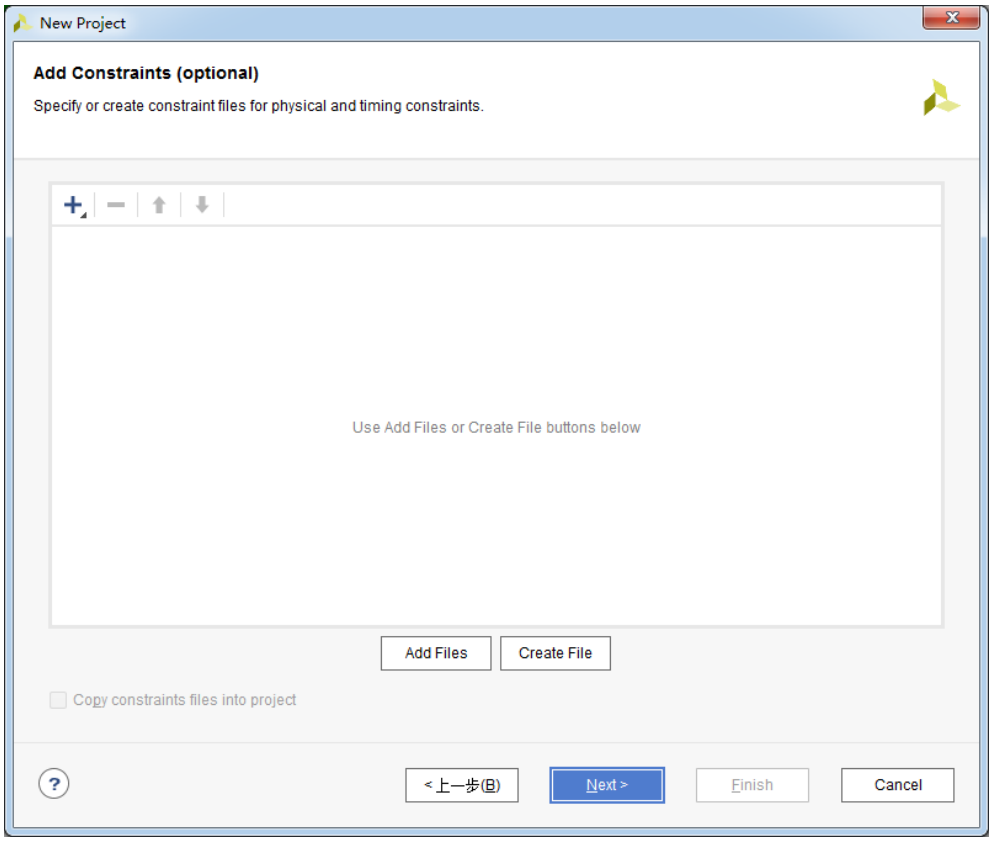
- 选择创建一个 RTL_Project，单击 Next



- 继续点击 Next



- 继续点击 Next


New Project
Add Constraints (optional)
Specify or create constraint files for physical and timing constraints.
Use Add Files or Create File buttons below
Add Files
Create File
Copy constraints files into project
< 上一步(B)
Next >
Finish
Cancel

- 在弹出的对话框中：

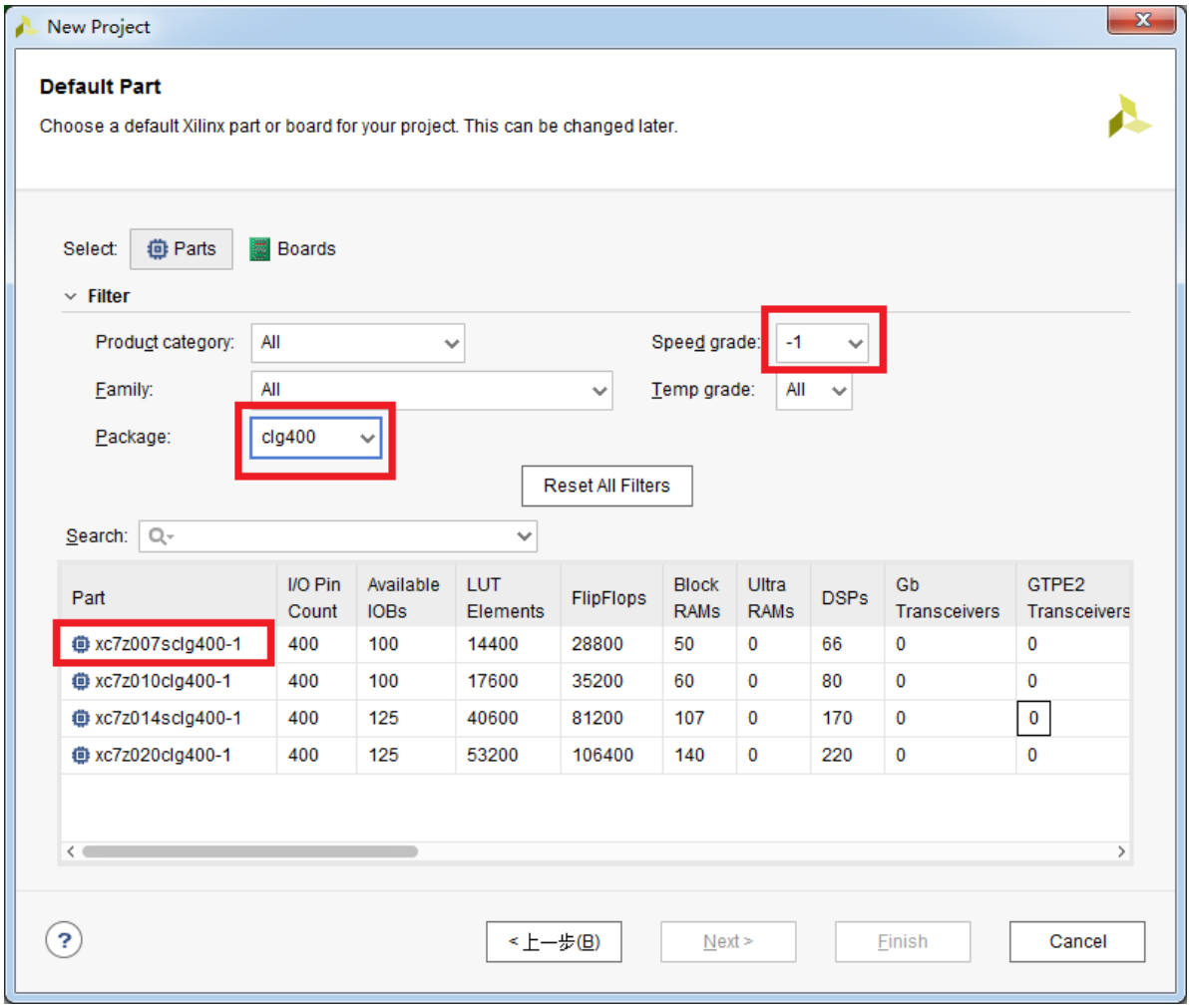
a. 单击选择 Parts。

b. 点击下拉框将 speed grade 设置为-1。

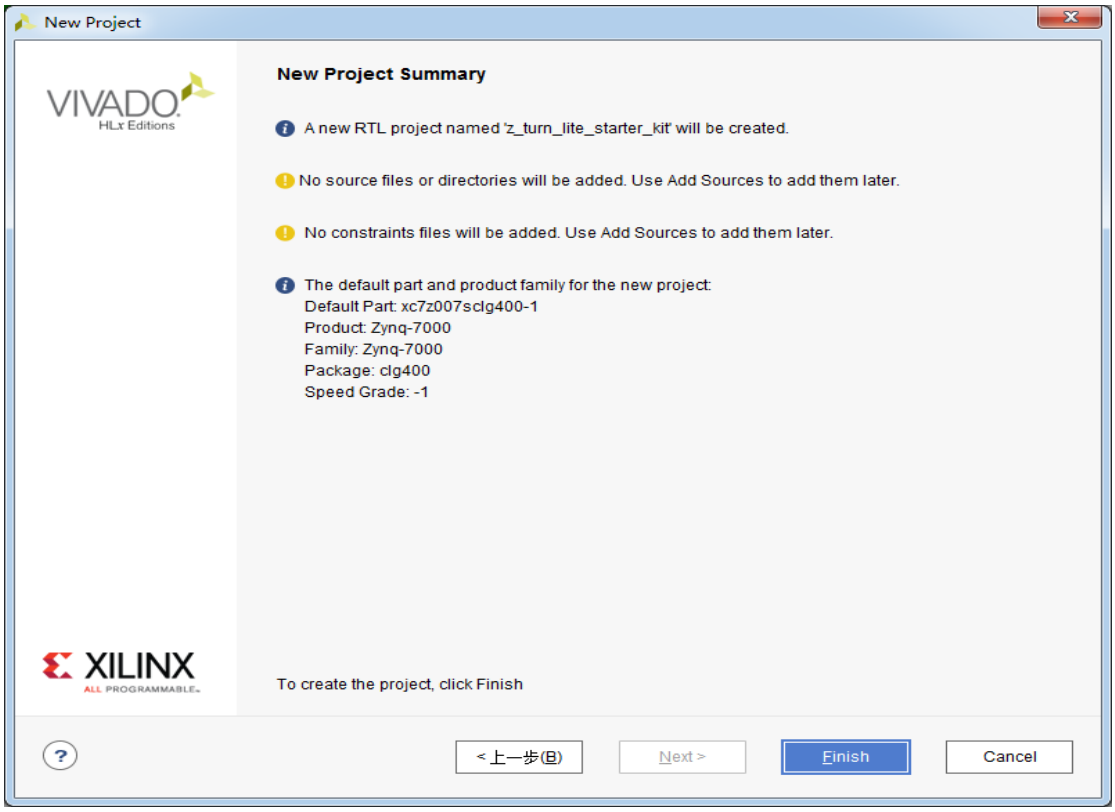
c. 封装为设置为 clg400。

d. 选择开发板芯片为 xc7z007sclg400-1(如果你的开发板型号为 7z010,请选择芯片型号为 xc7z010clg400-1)。

e 点击 Next。

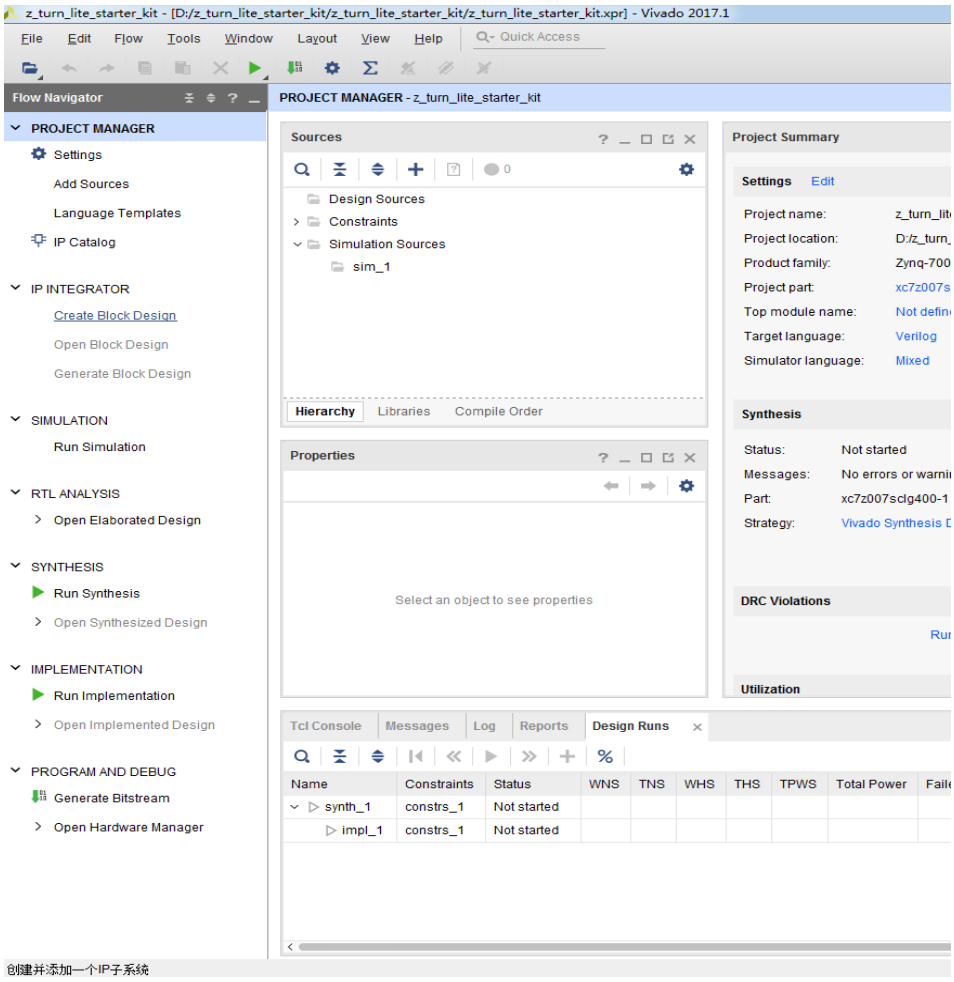


- 在这个新建工程对话框中，单击 Finish

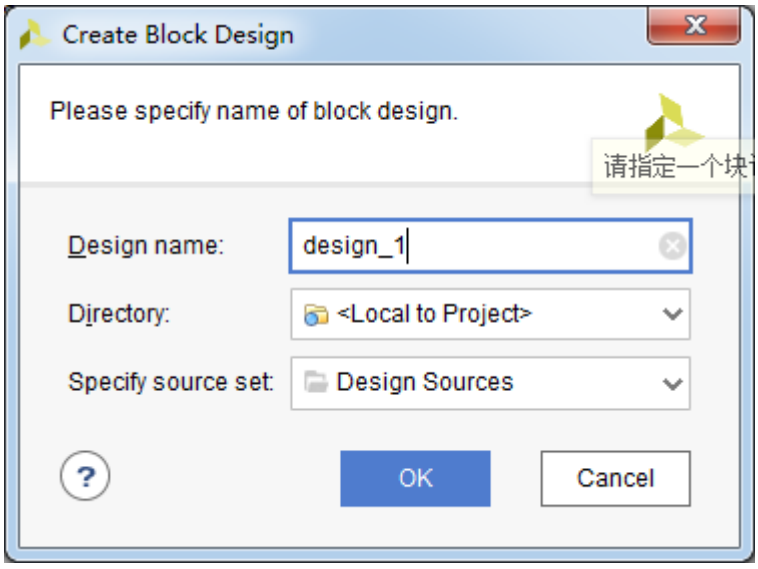


4.1 创建一个 Block Design

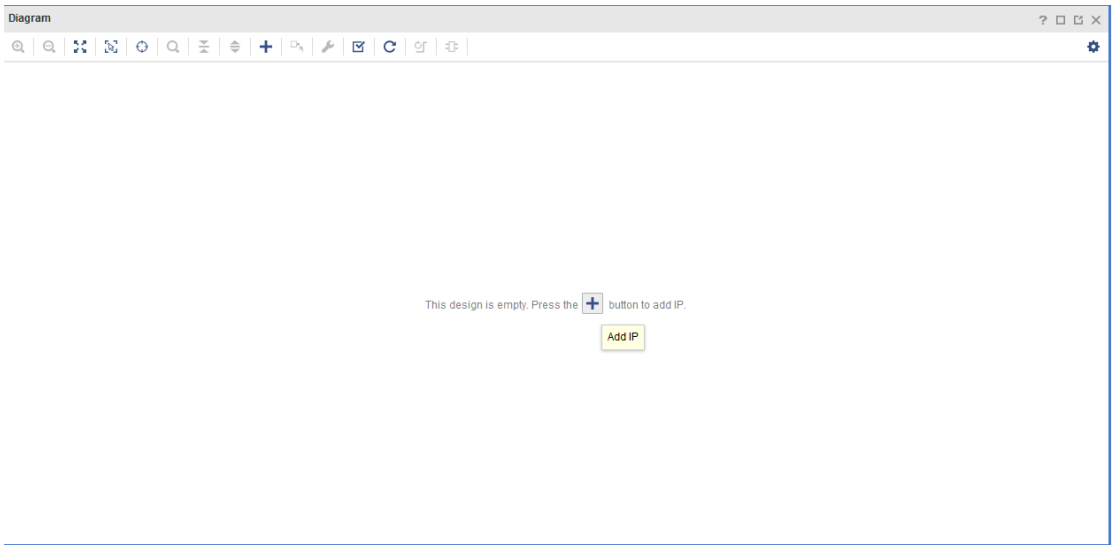
•单击 IP Integrator->Create Block Design 新建一个 Block Design。



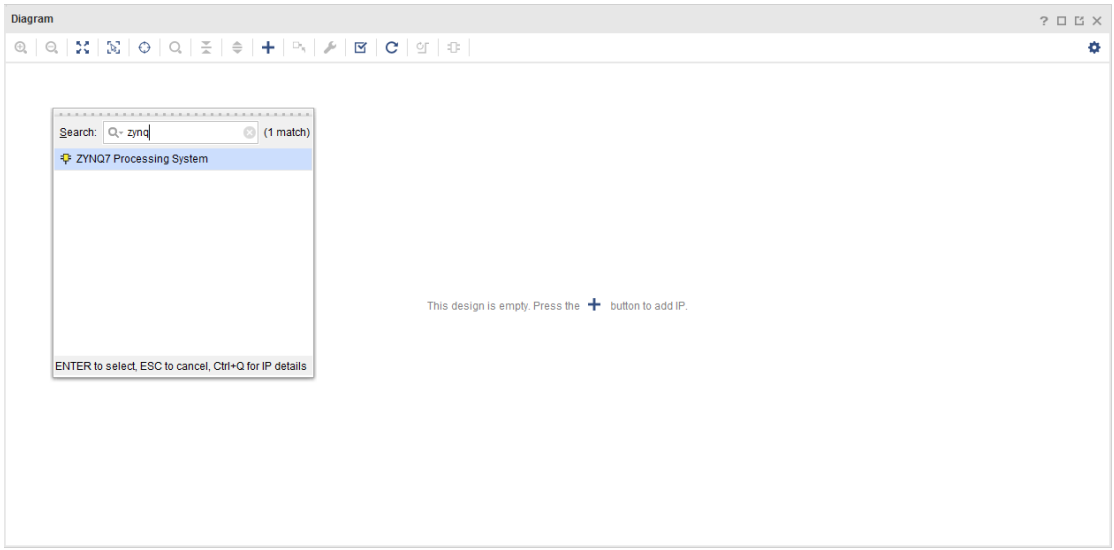
•你可以单击 Design name 可以将 design_1 改为你想要的名称，这里将 design_1 作为默认名称不作更改



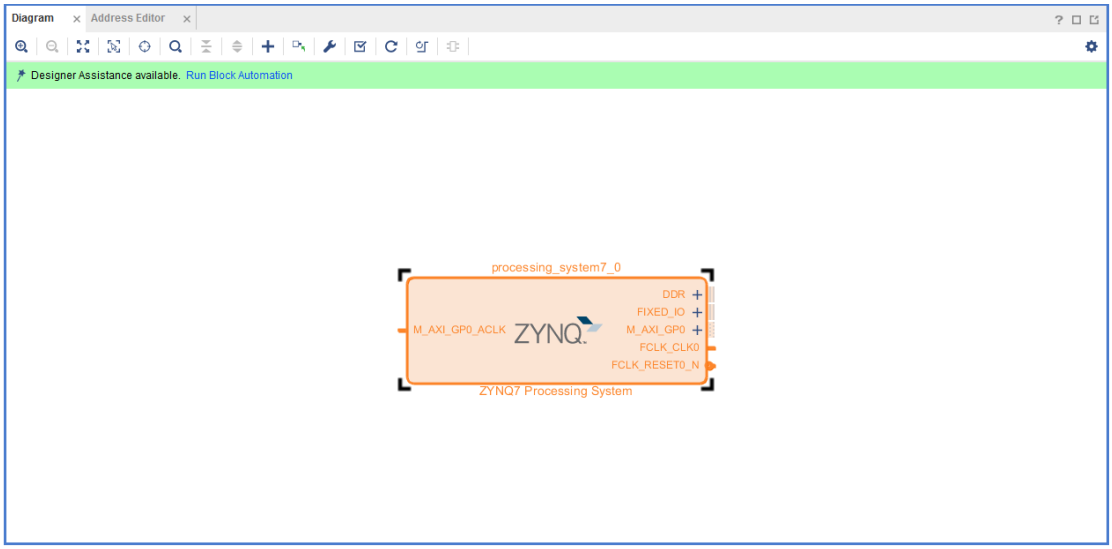
•点击下图中的 Add IP 添加一个 IP 核



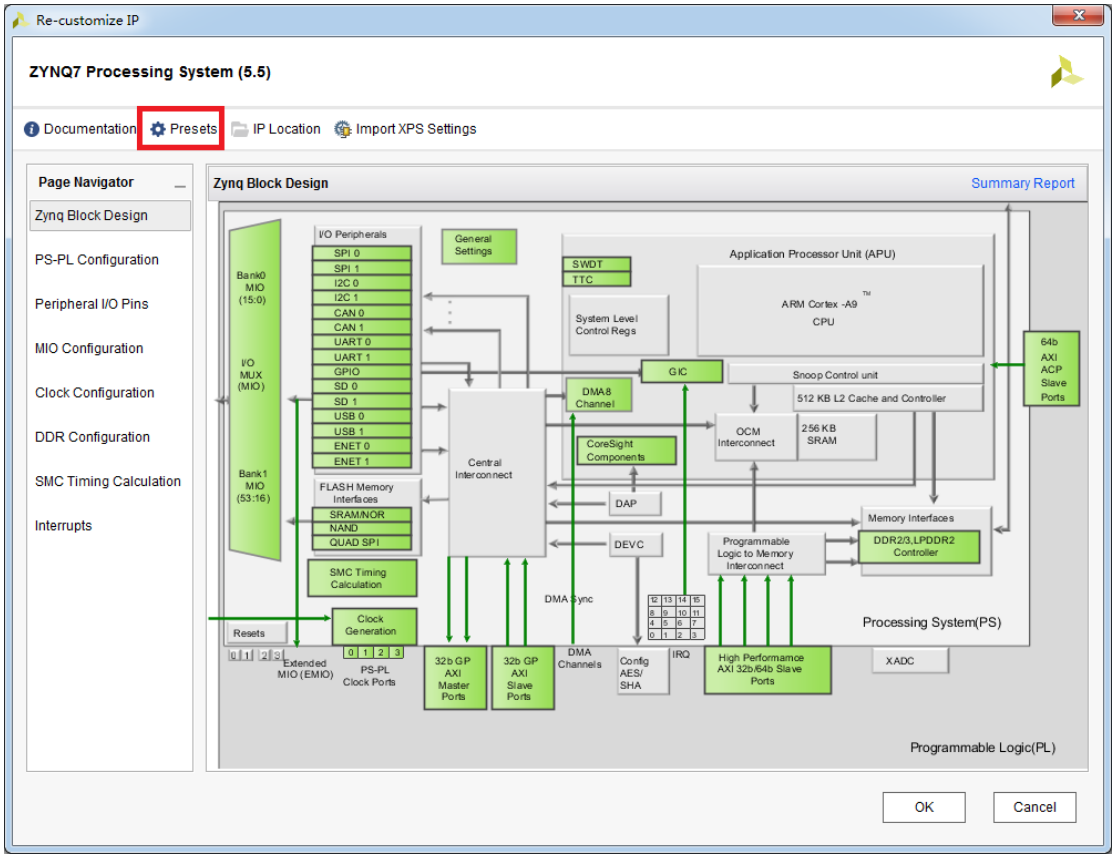
•点击搜索栏输入 ZYNQ，然后双击搜索到的 ZYNQ 核



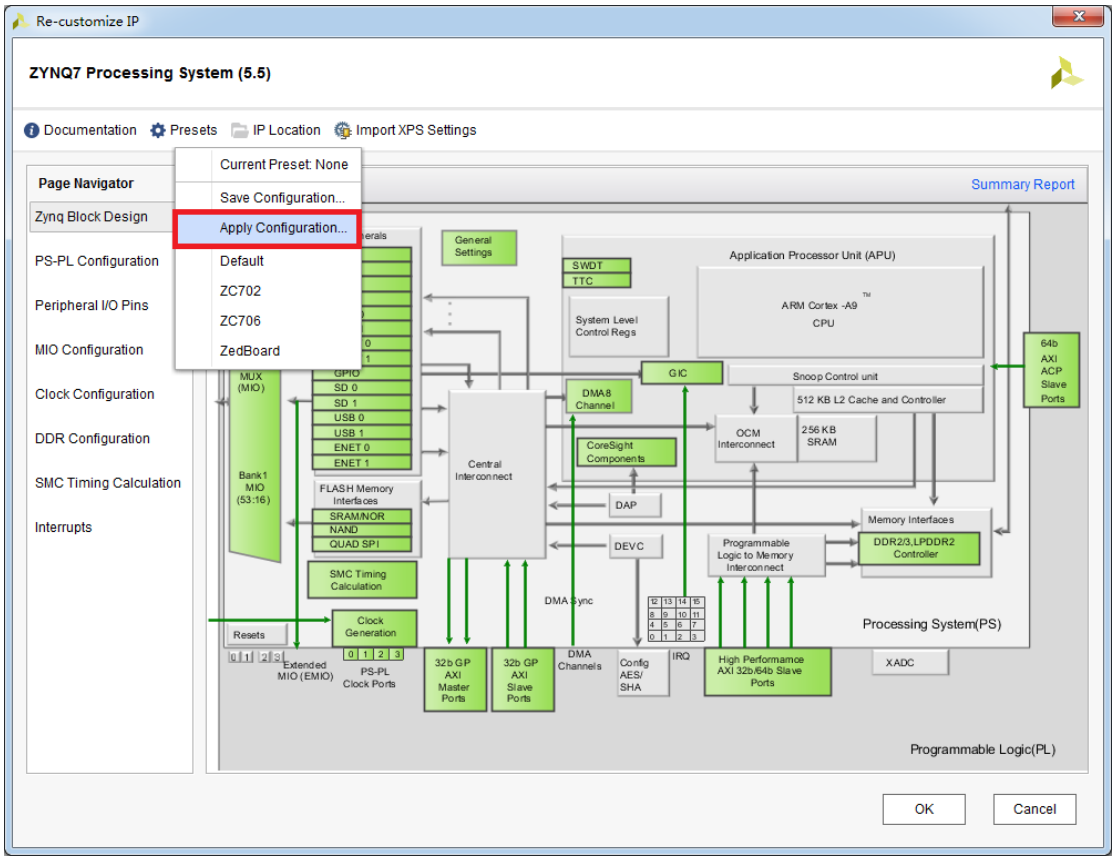
•ZYNQ IP 核被添加到设计中，如下图所示



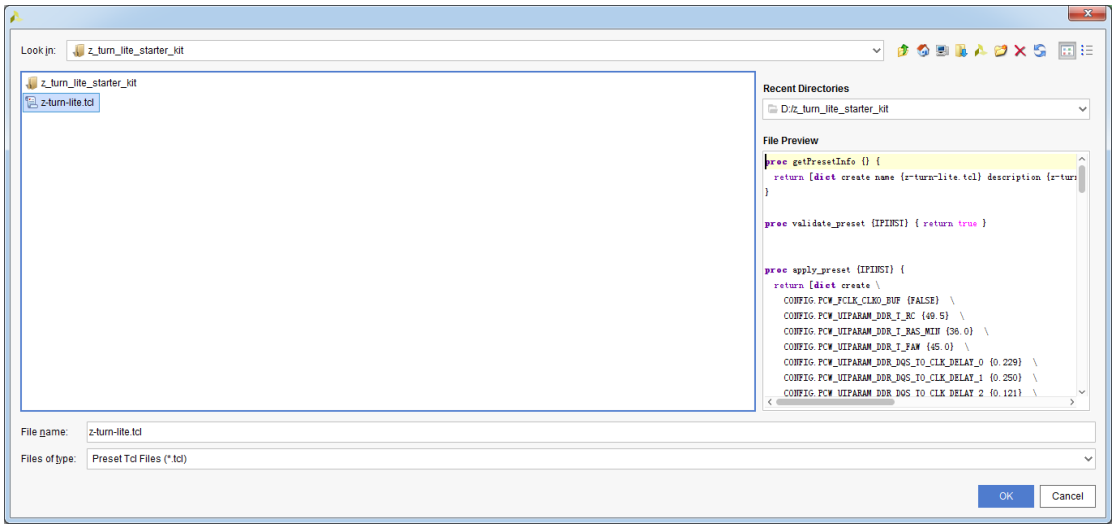
•双击 ZYNQ IP 核，然后单击 Presets



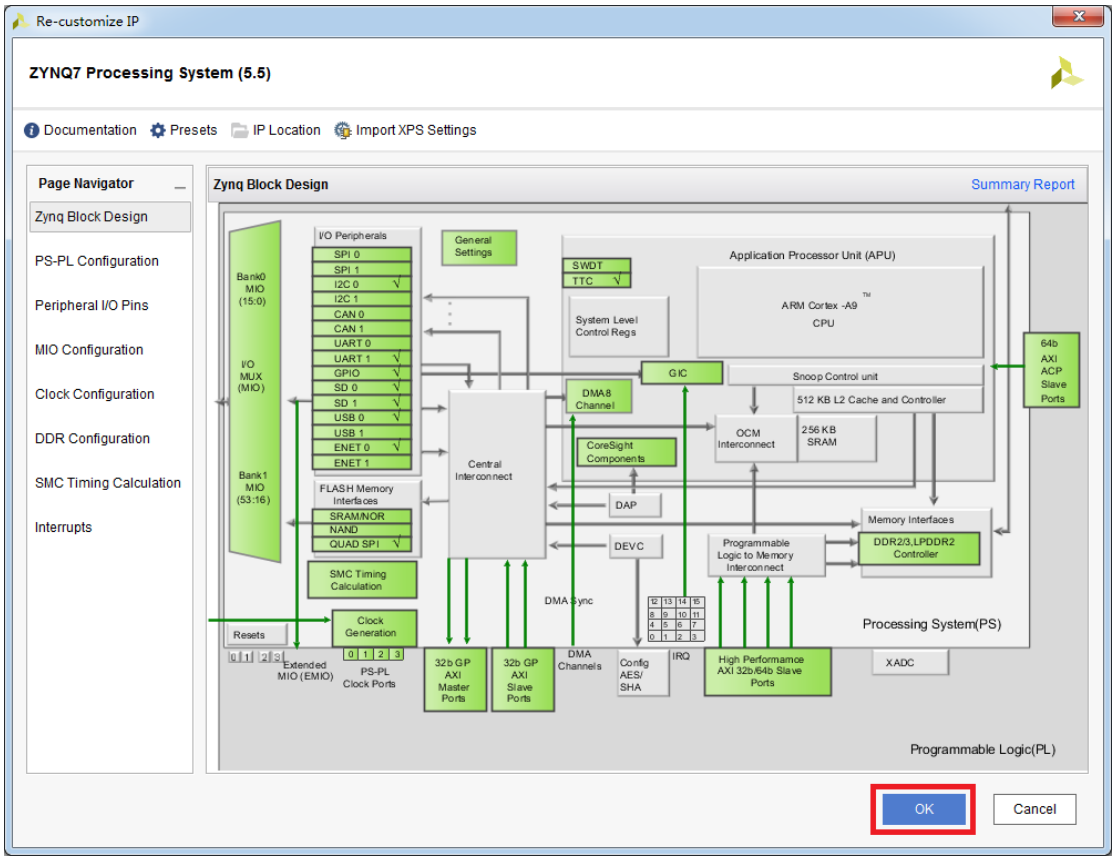
•选择 Applying configuration 添加 ps 核的配置文件



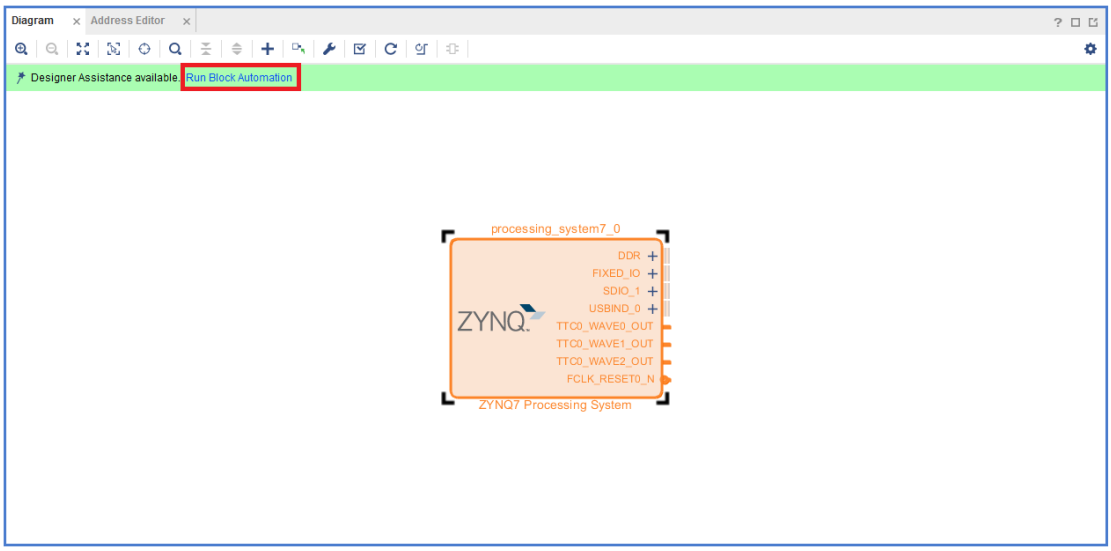
•选中我们所提供的 z-turn-lite.tcl 文件，然后点击 OK



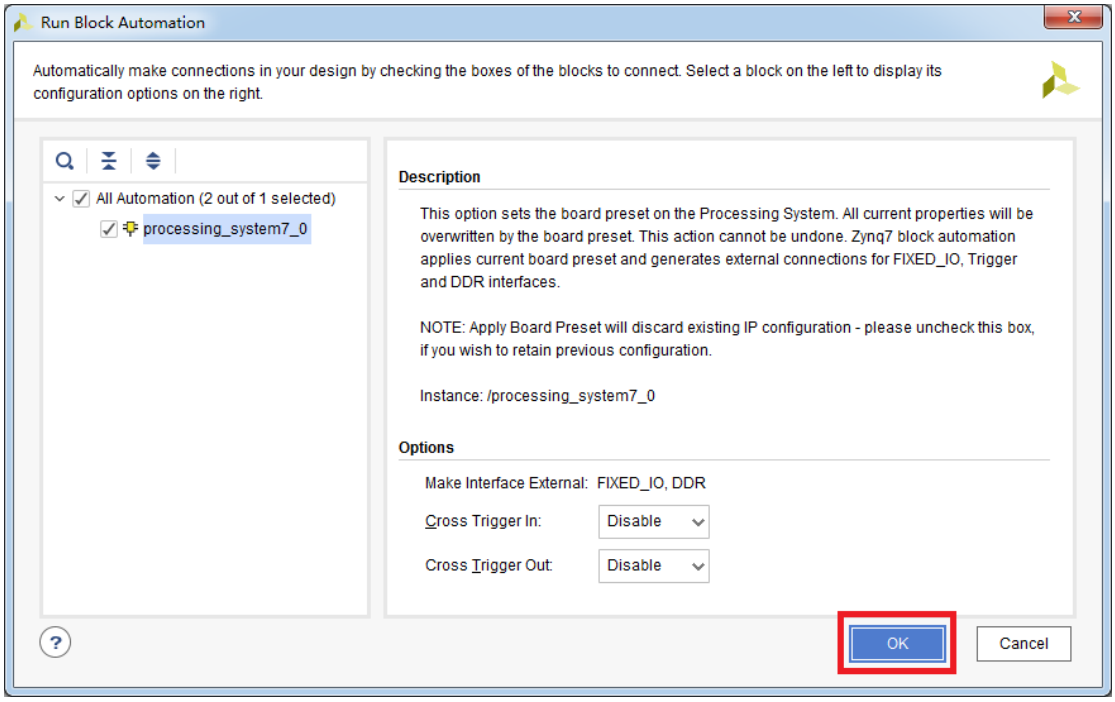
•继续点击 OK



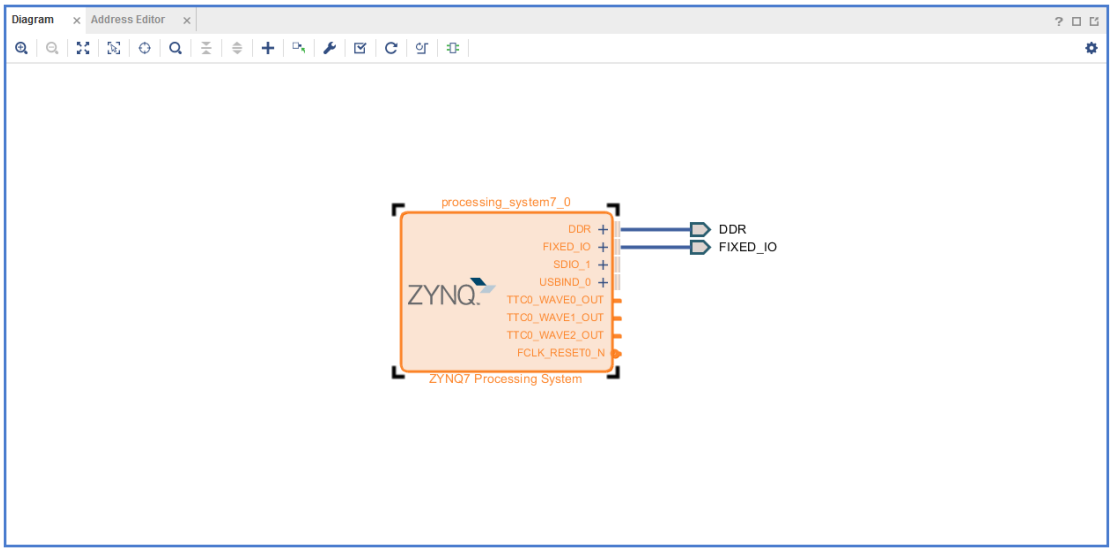
•配置完成后单击 Run Block Automation 进行自动连线



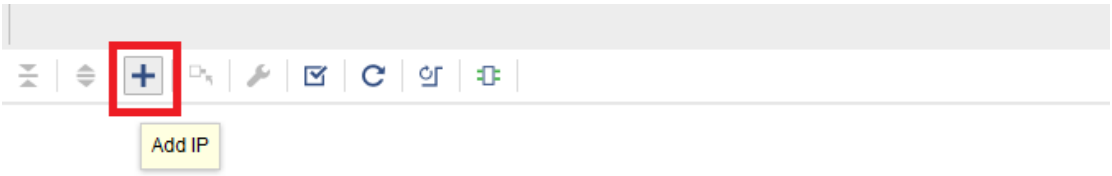
•在弹出的对话框中，点击 OK

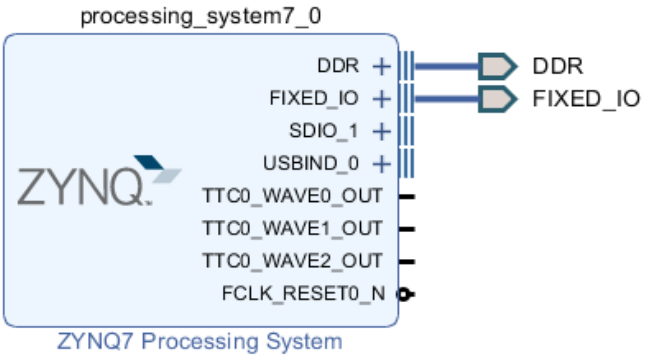


• ZYNQ IP 核的配置如下图，可以看到 ZYNQ IP 核配置了 DDR、PS 外设、时钟等。

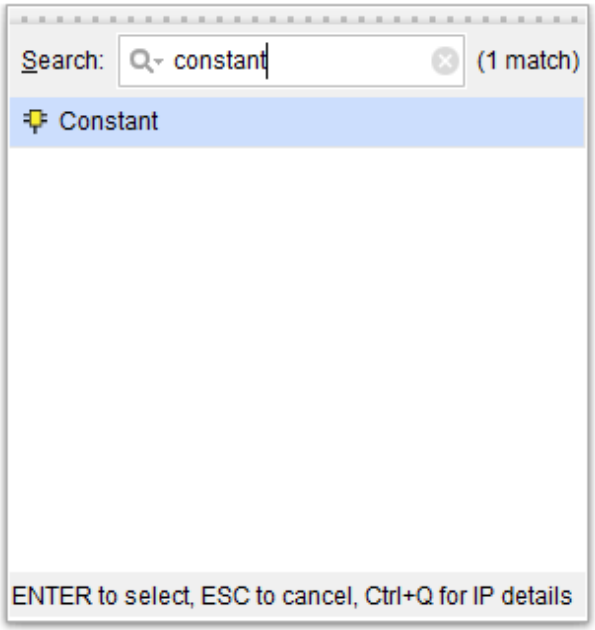


单击工具栏上 Add IP 添加一个 IP 核

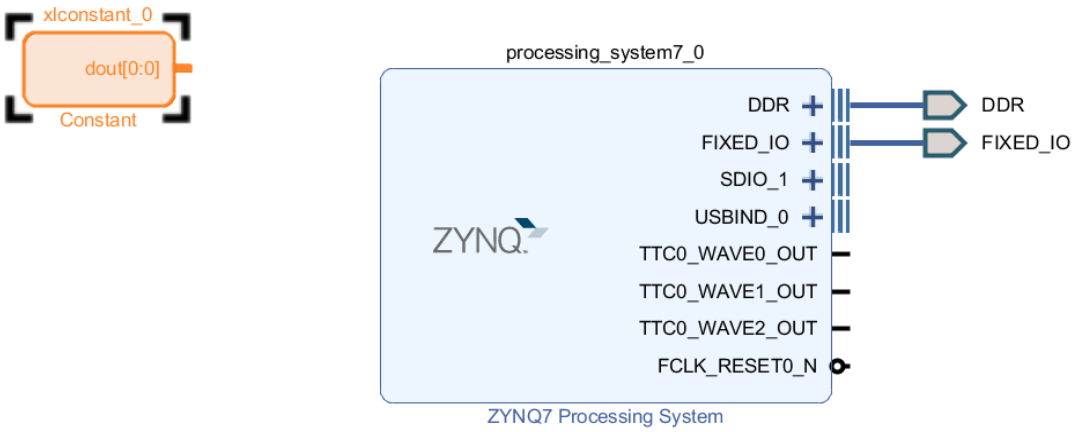




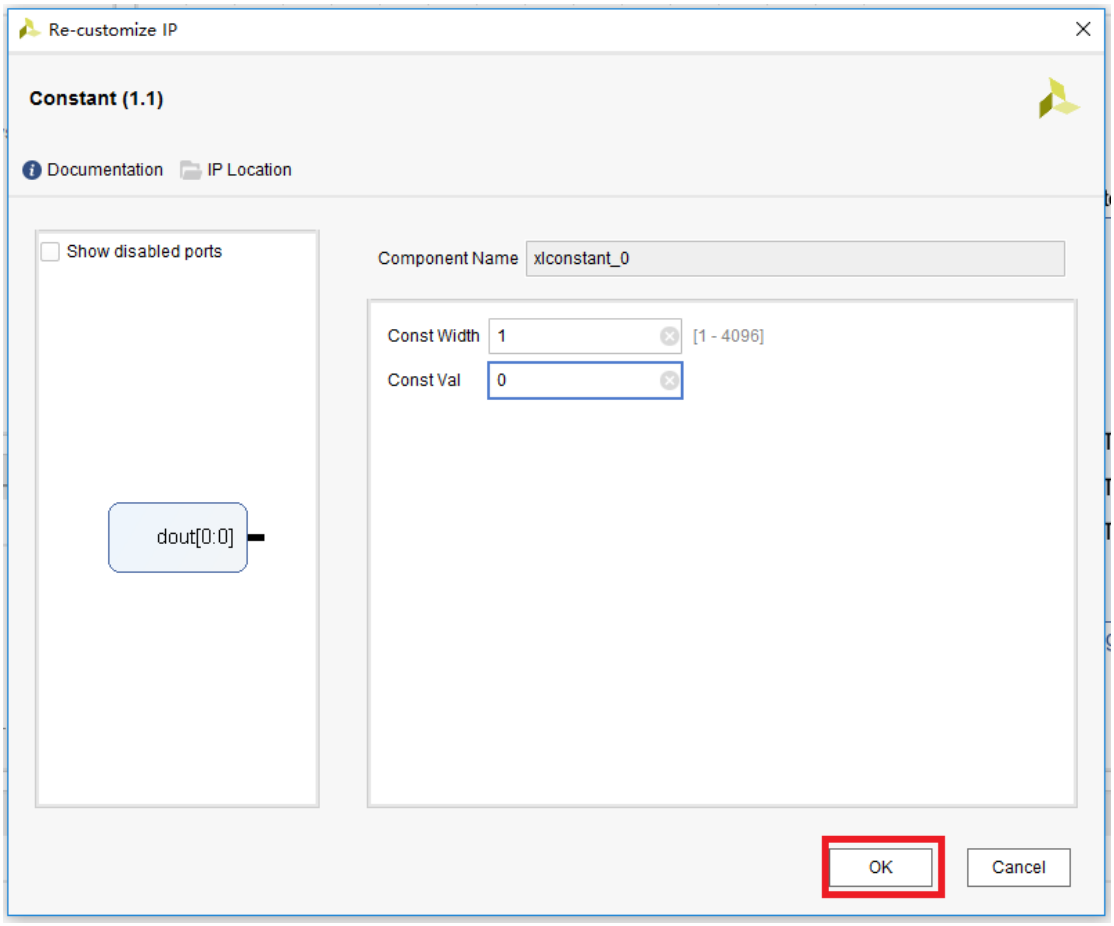
在弹出的搜索栏中输入 constant，然后双击搜索到 constant IP 核



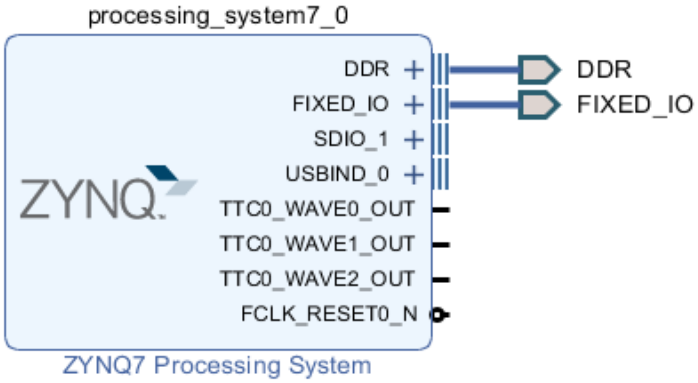
将 constant IP 核添加到设计中如下图所示

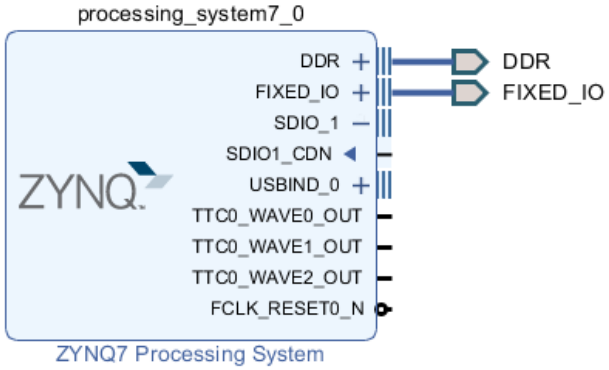


双击 Constant IP 核，对 IP 核进行配置将 Const Val 设置为 0，然后单击 OK

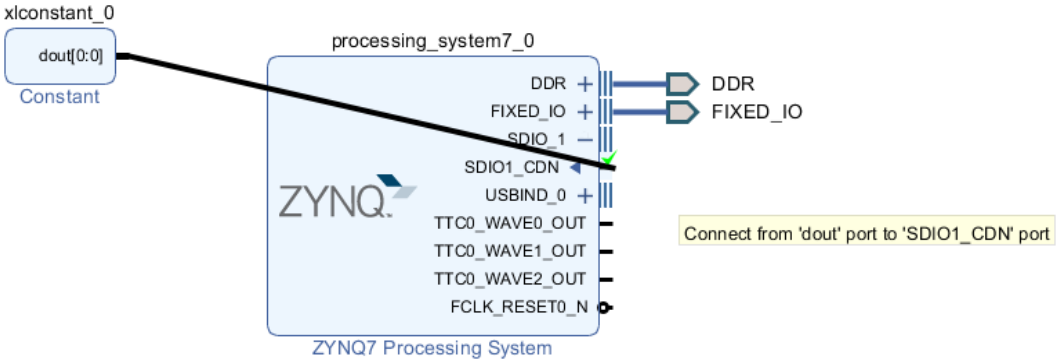


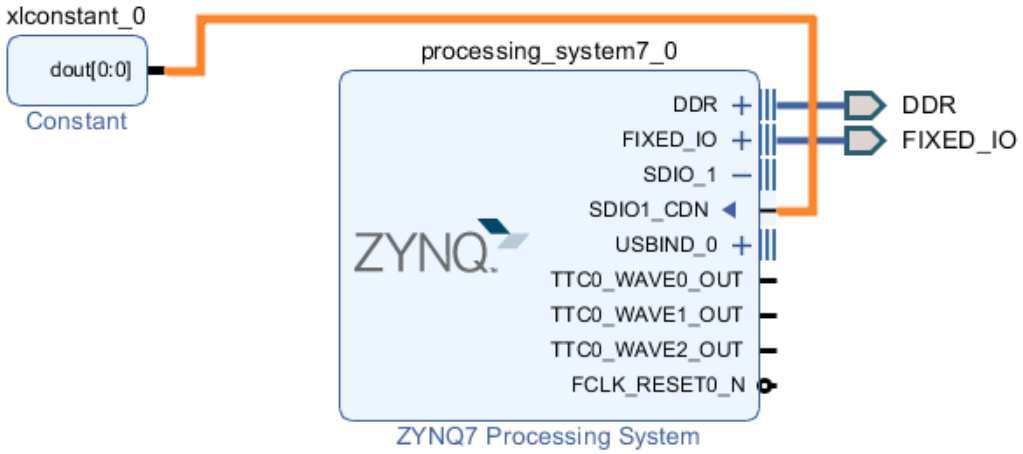
将 SDIO_1 的 +号点开，展开 SDIO_1 的引脚



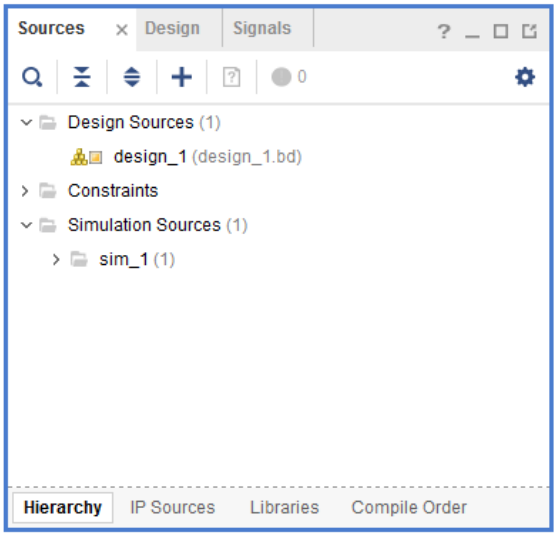


用鼠标点击 SDIO1_CDN 待鼠标变成铅笔后将引脚拖到 dout 的引脚上如下图所示

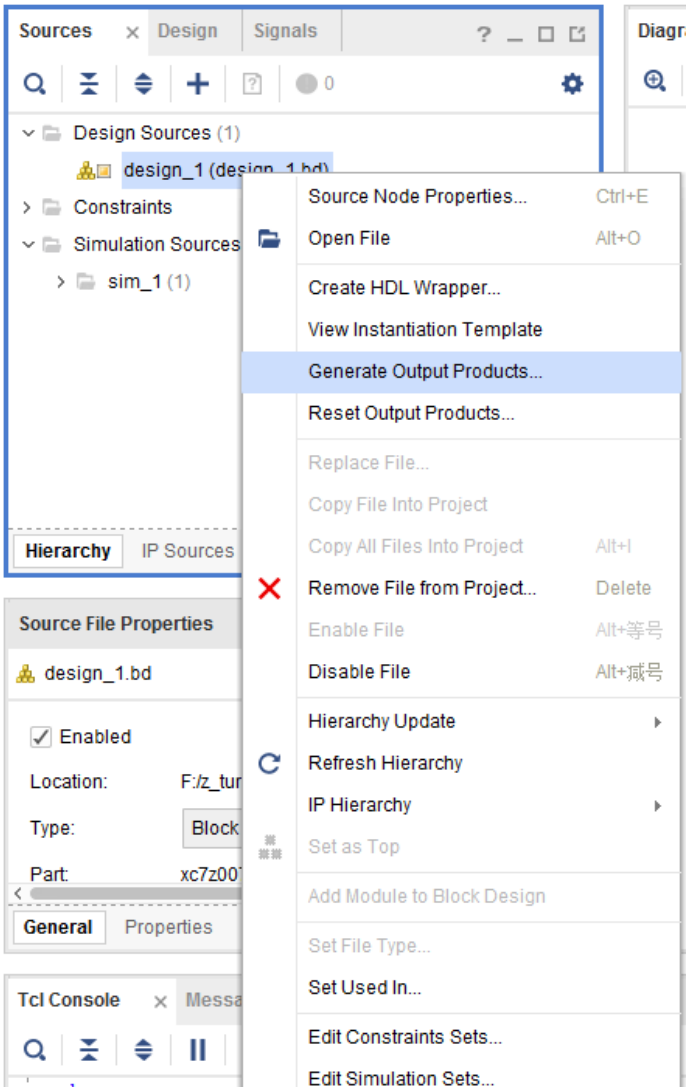




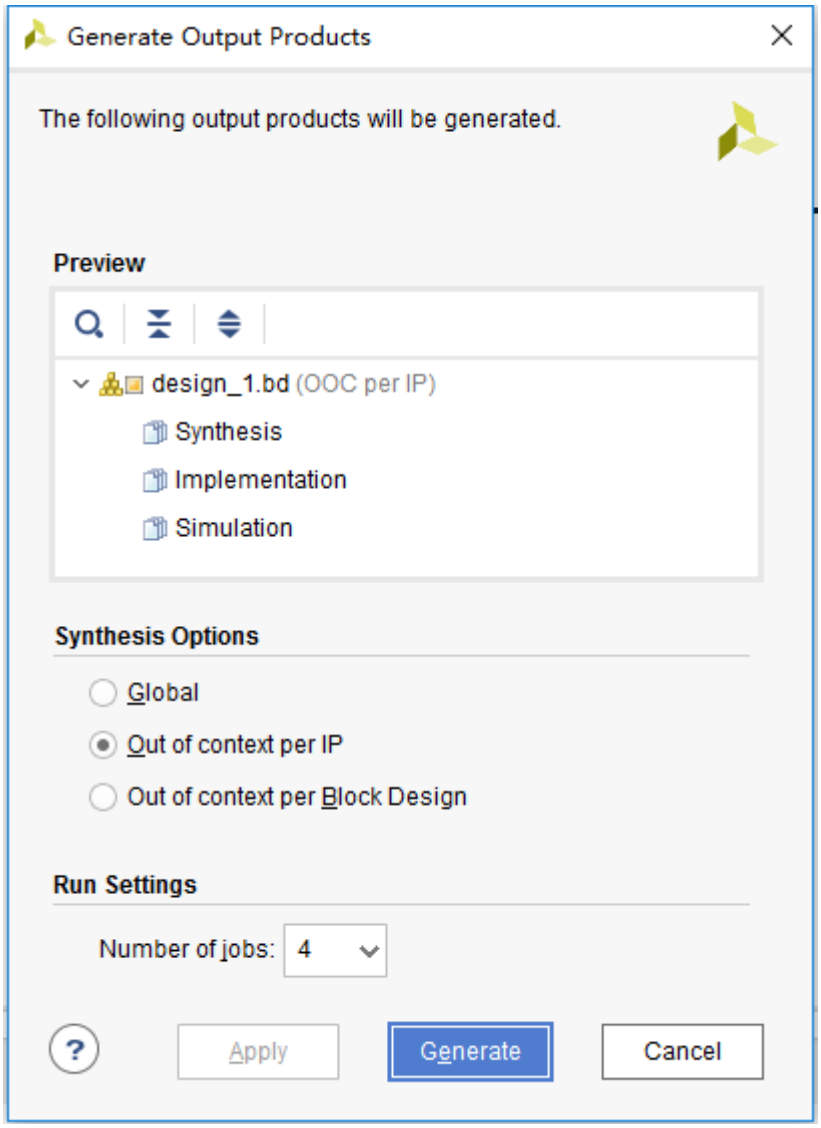
单击 sources，如下图所示



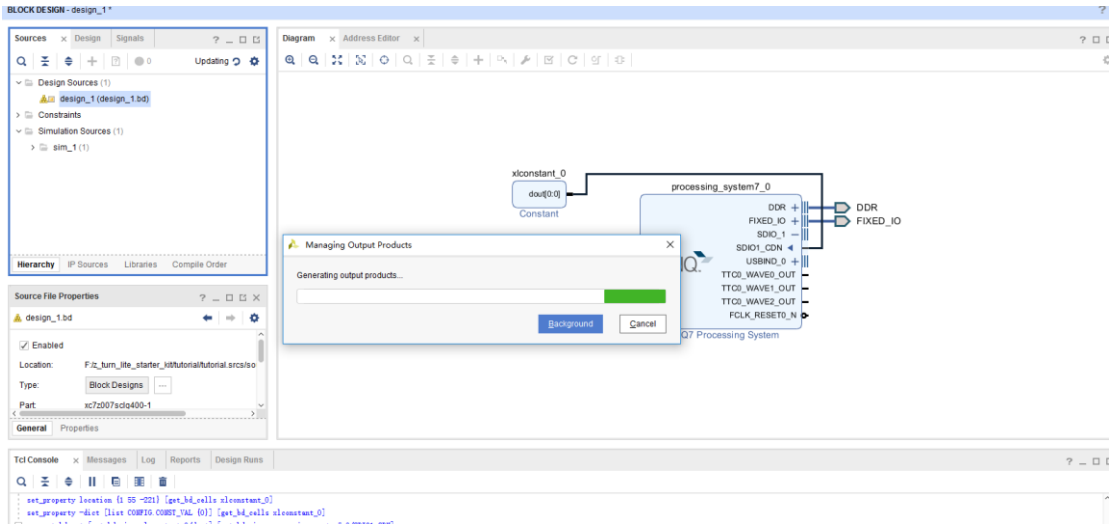
右击 **Generate Output Products**



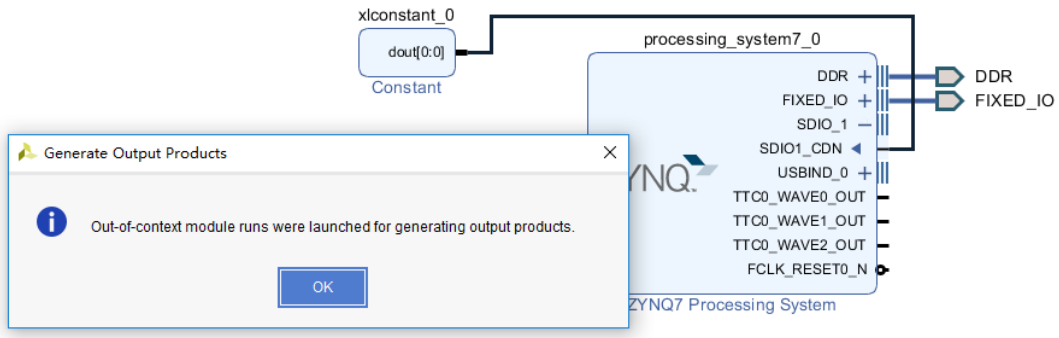
然后单击 **Generate**



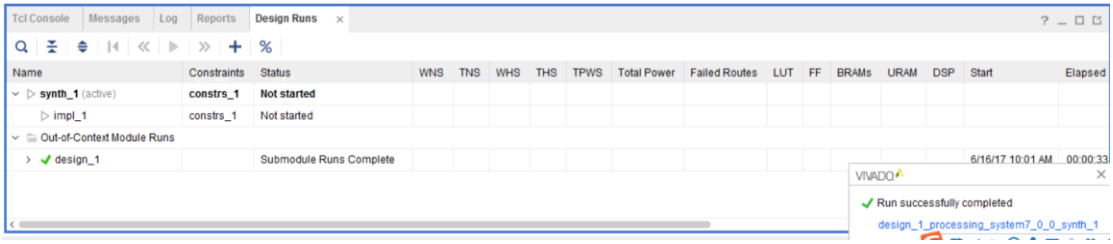
正在生成中



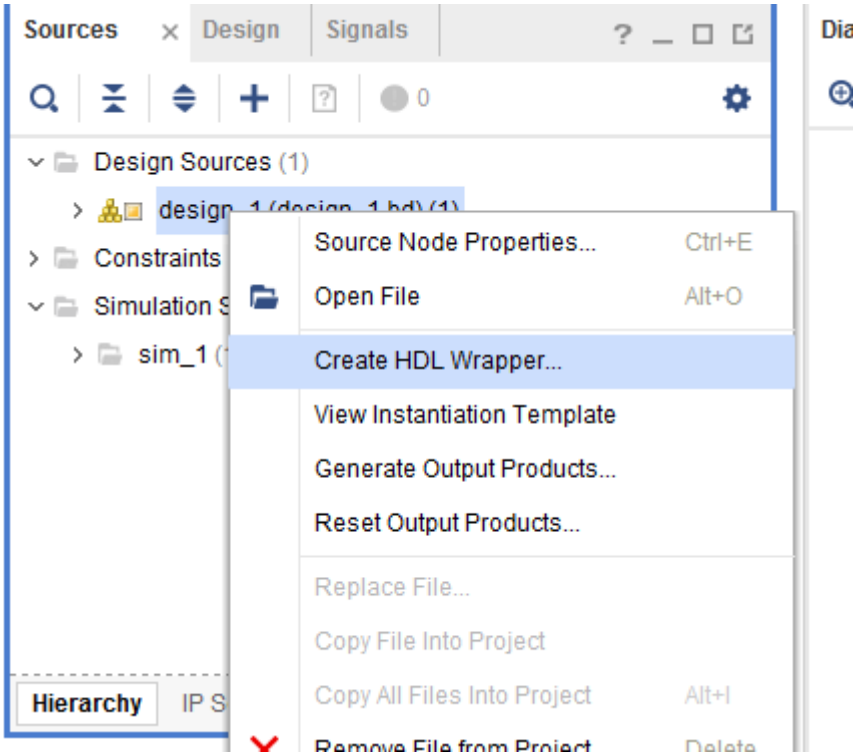
单击 OK



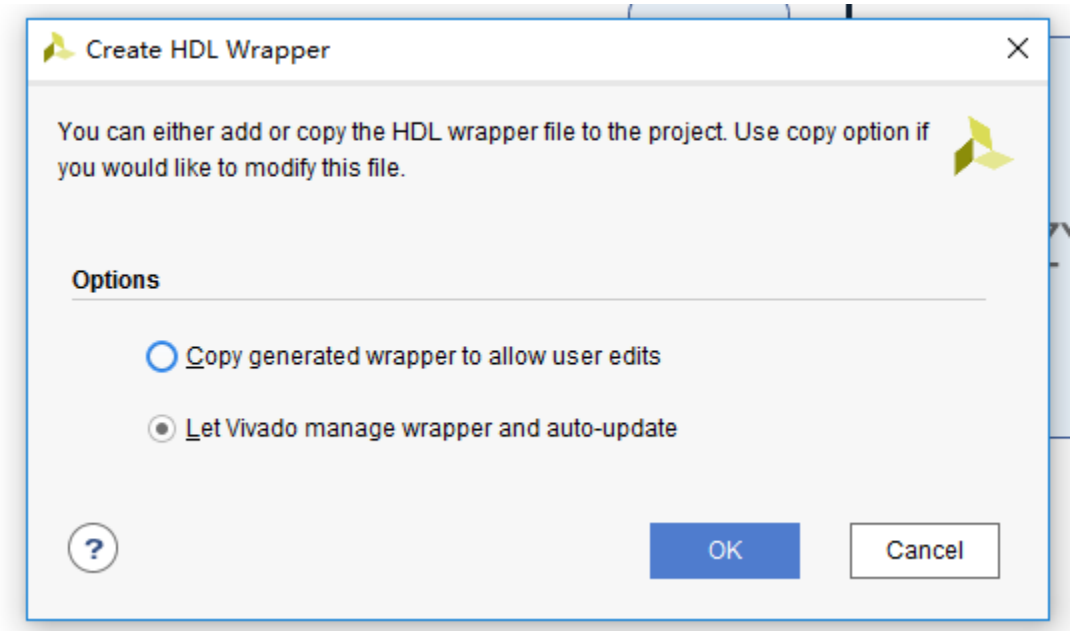
如下图所示，生成综合文件成功



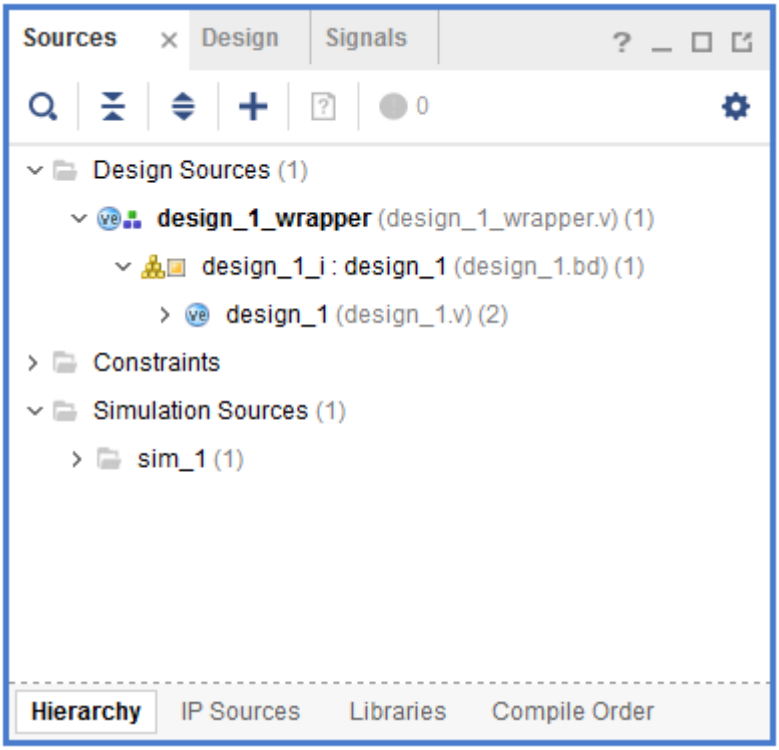
右击 **design_1** 然后选择 **Create HDL Wrapper** 生成顶层文件



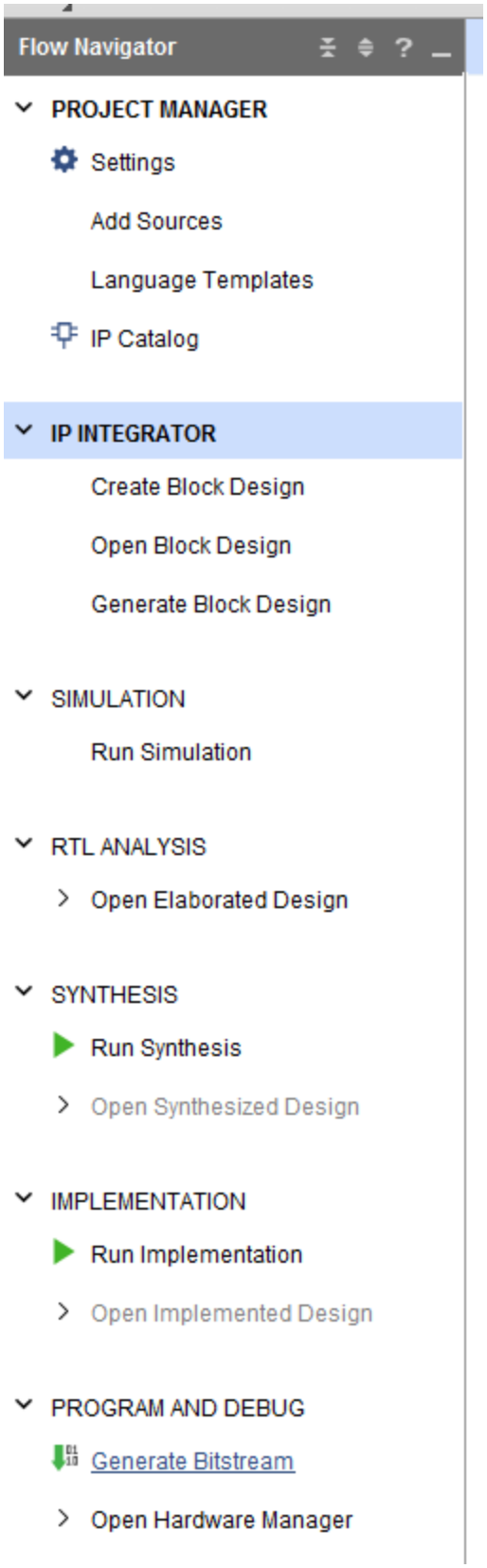
单击 OK



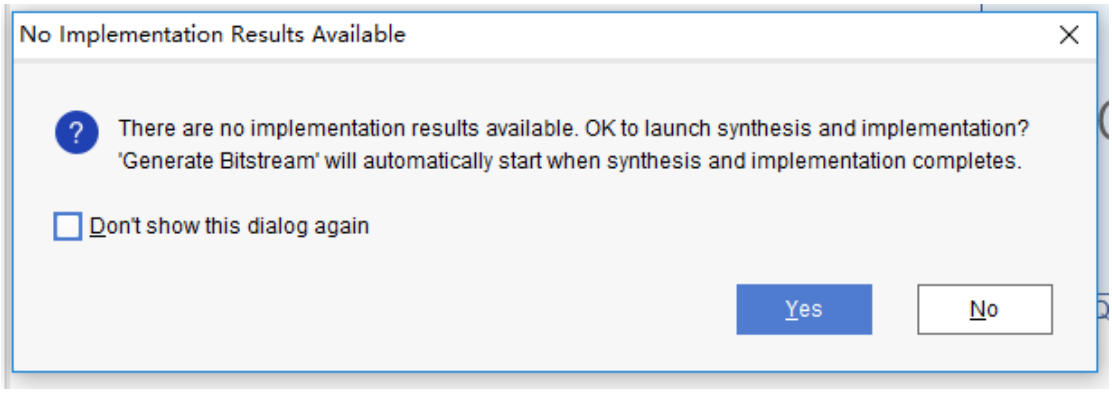
自动生成顶层文件 design_1_warpper.v



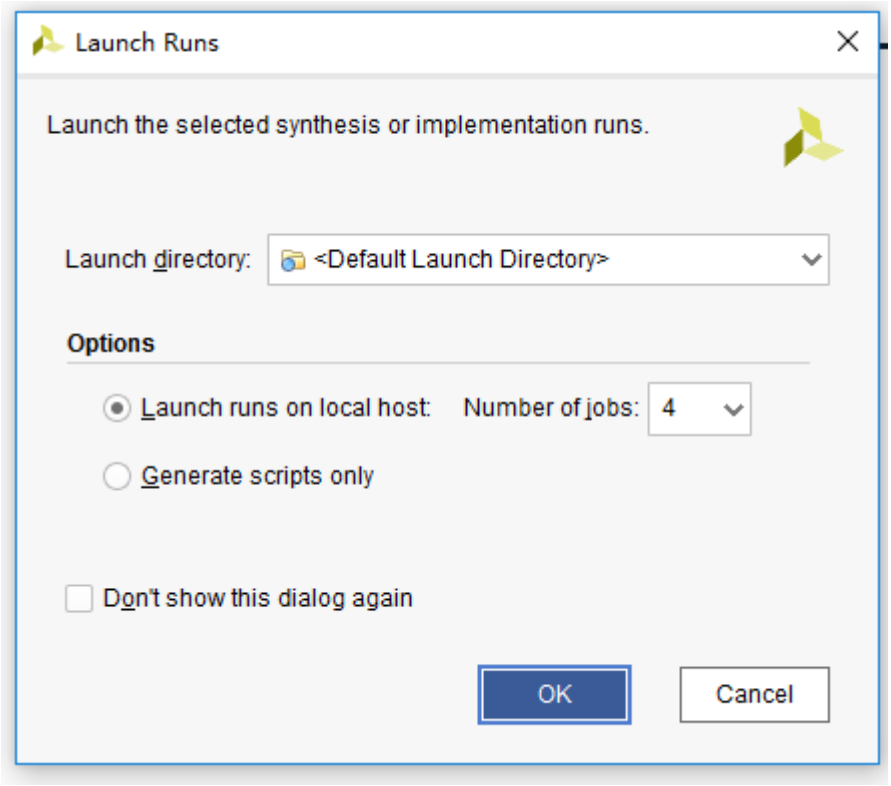
单击 **Generating Bitstream** 生成二进制文件



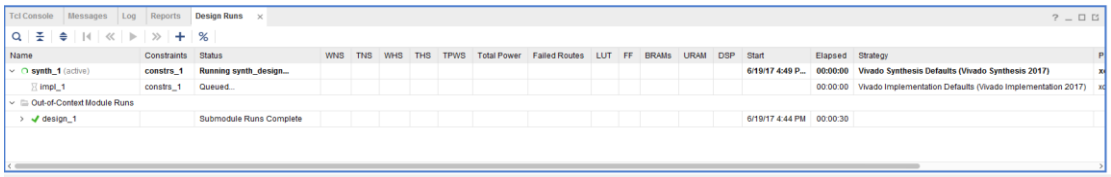
在出现对话框中点击 Yes



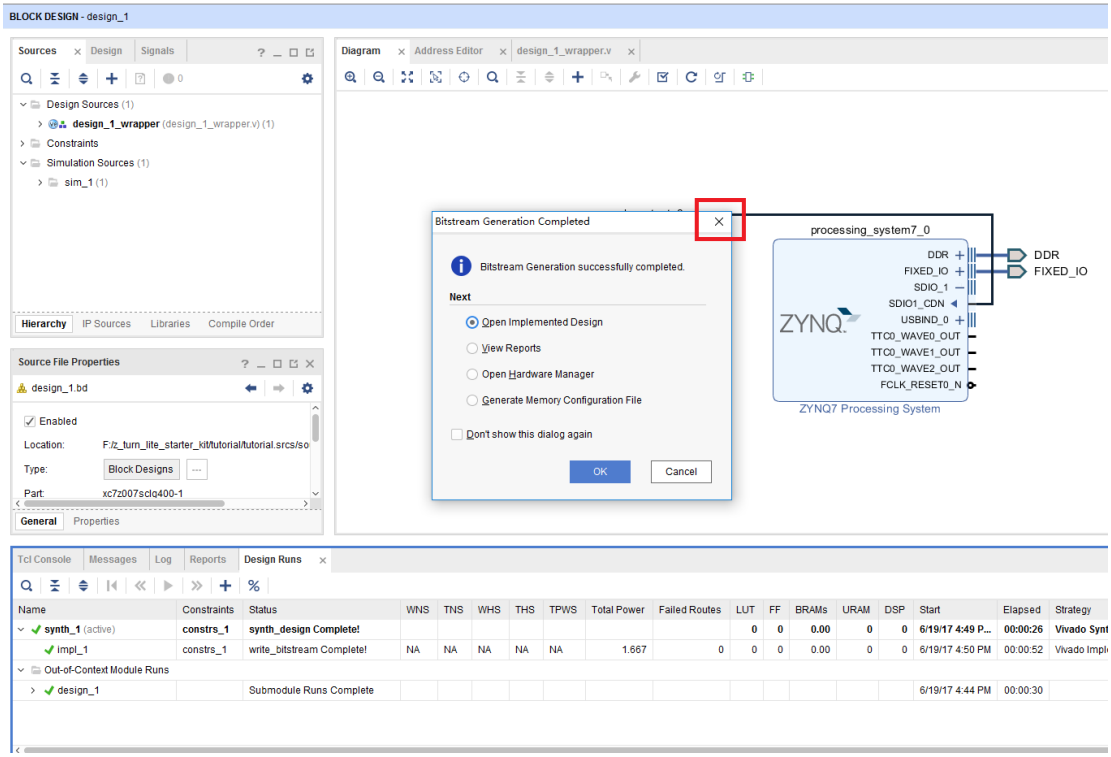
单击 OK



正在生成二进制文件 bit

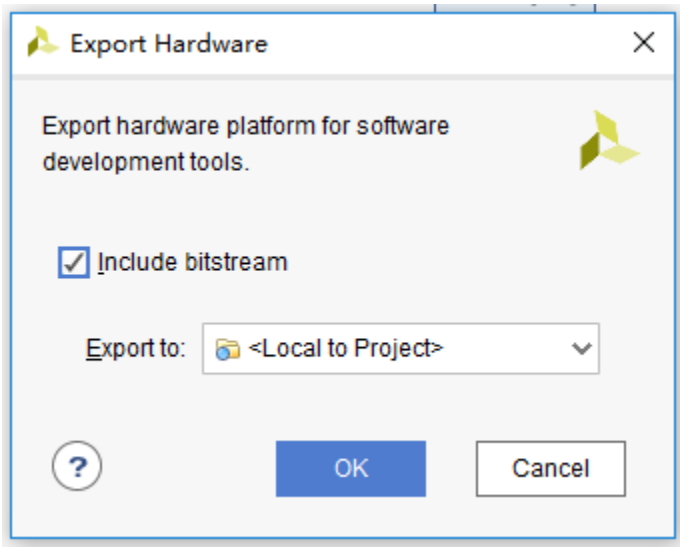


二进制文件 bit 生成完成，关闭弹出的对话框

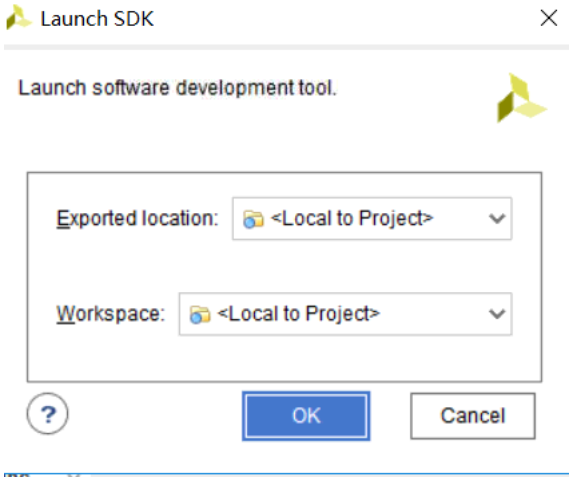


5 将硬件平台导出到 SDK

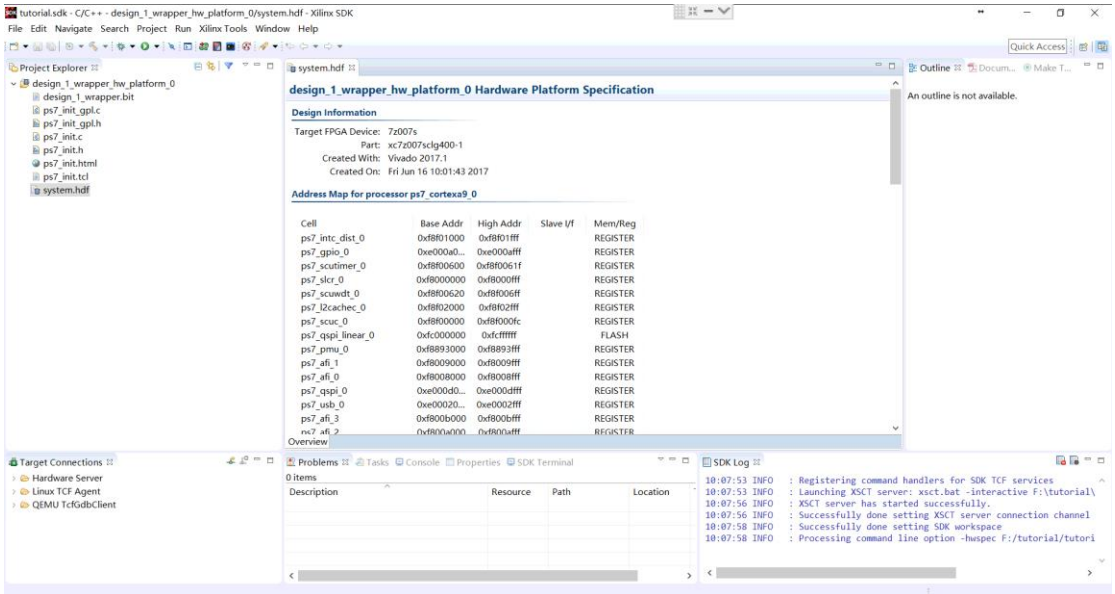
选择 File > Export > Export Hardware，勾选 Include bitstream，然后点击 OK



点击菜单栏上面的 File > Launch SDK 启动 SDK，然后单击 OK



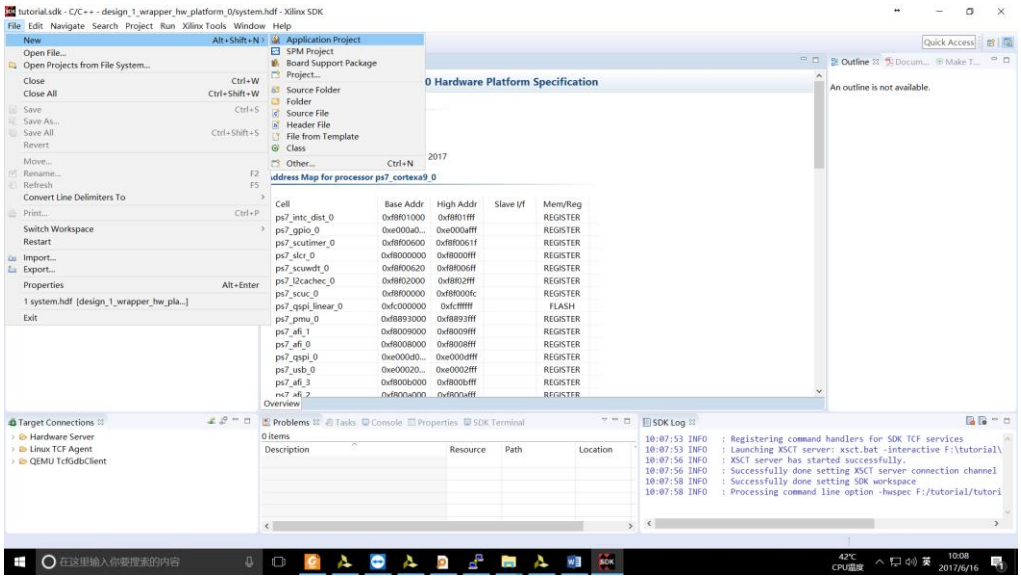
硬件平台导入后打开 SDK，在下图所示的 design_1_wrapper_hw_platform_0 将用于生成硬件平台的 BSP。



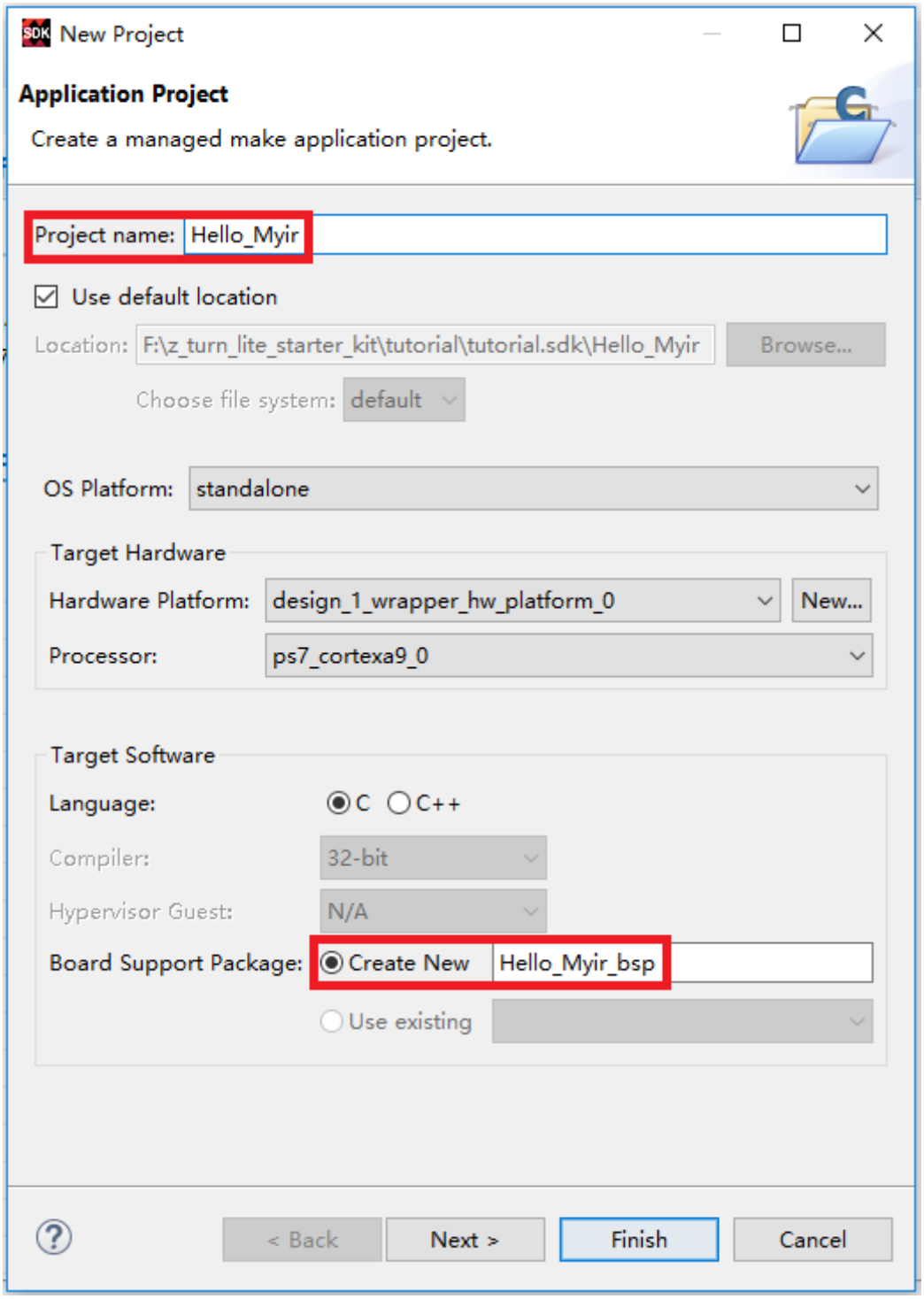
5.1 创建一个“Hello World”工程

我们可以直接创建软件应用程序，因为 SDK 会自动为我们生成一个 BSP。我们将首先创建一个简单的“Hello World”应用程序，这样您就可以熟悉 SDK。

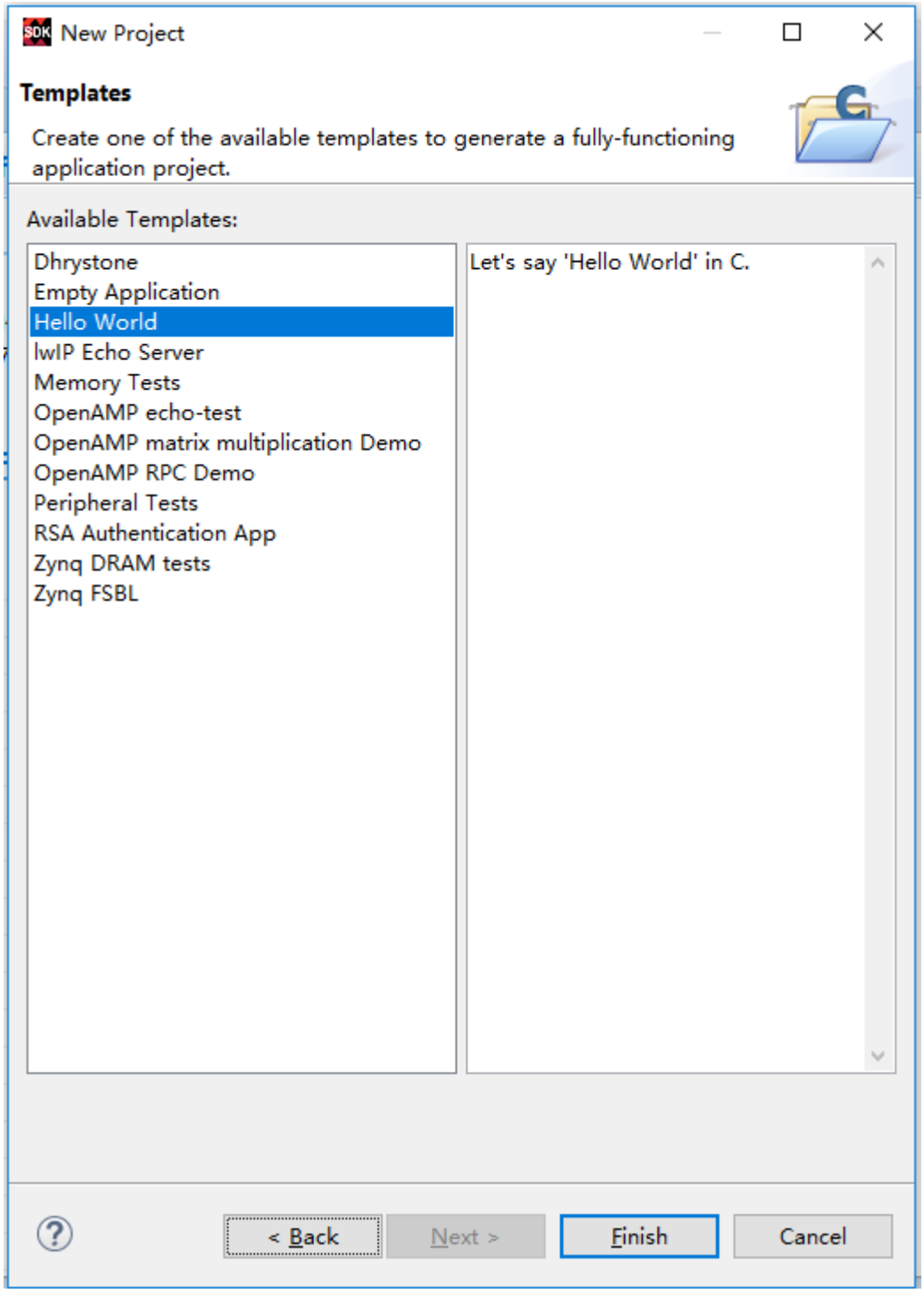
从 SDK 工具栏中，选择文件>新建应用程序项目来创建一个新的应用程序。



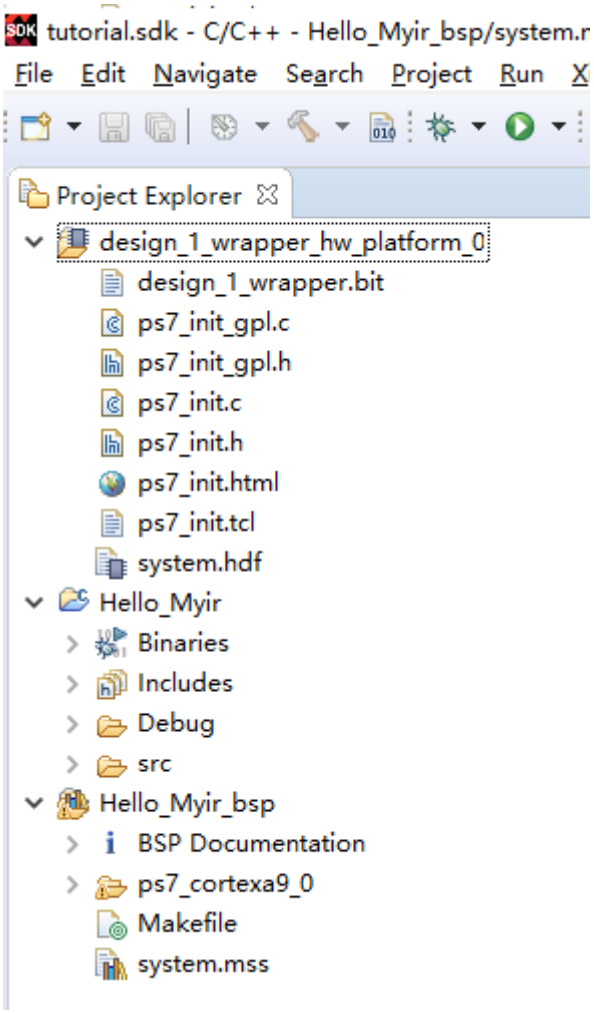
新建一个工程，工程名为 Hello_Myir，在这个开发板支持的 BSP 包选择 **Create new**，其它的都保持不变单击 Next



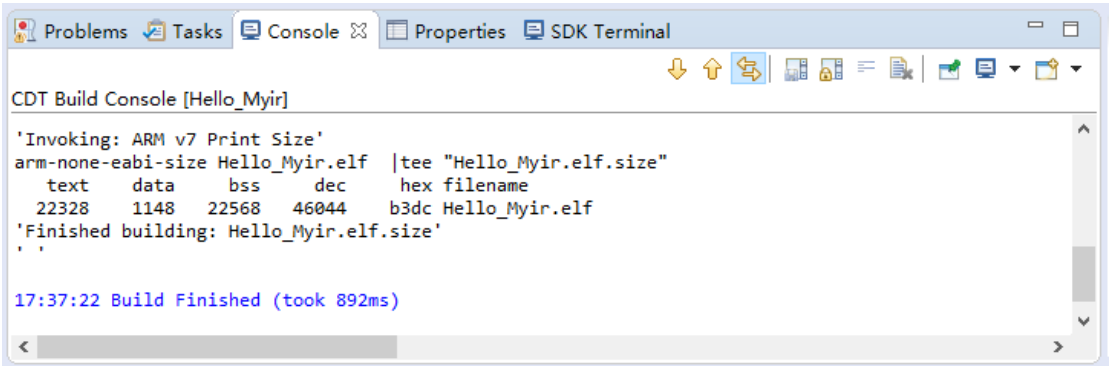
在弹出的对话框中选择 SDK 提供的模板 Hello World,然后单击 Finish



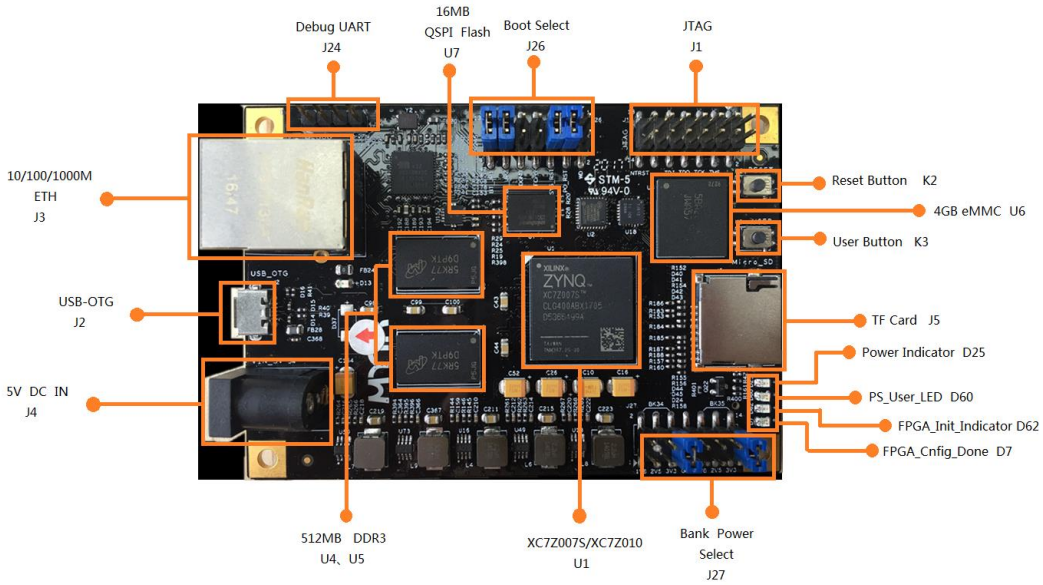
Hello_Myir 这个工程就添加到 SDK 中了，如下图所示。



将 Hello_World 这个工程编译完成后，在程序编译输出窗口可以看到生成了 Hello_Myir.elf，后面的章节将会介绍如何将 Hello_Myir.elf 文件加载到 Z-Turn-Lite DDR3 中运行



6 设置硬件



请按照以下步骤设置 z_turn_Lite 开发板

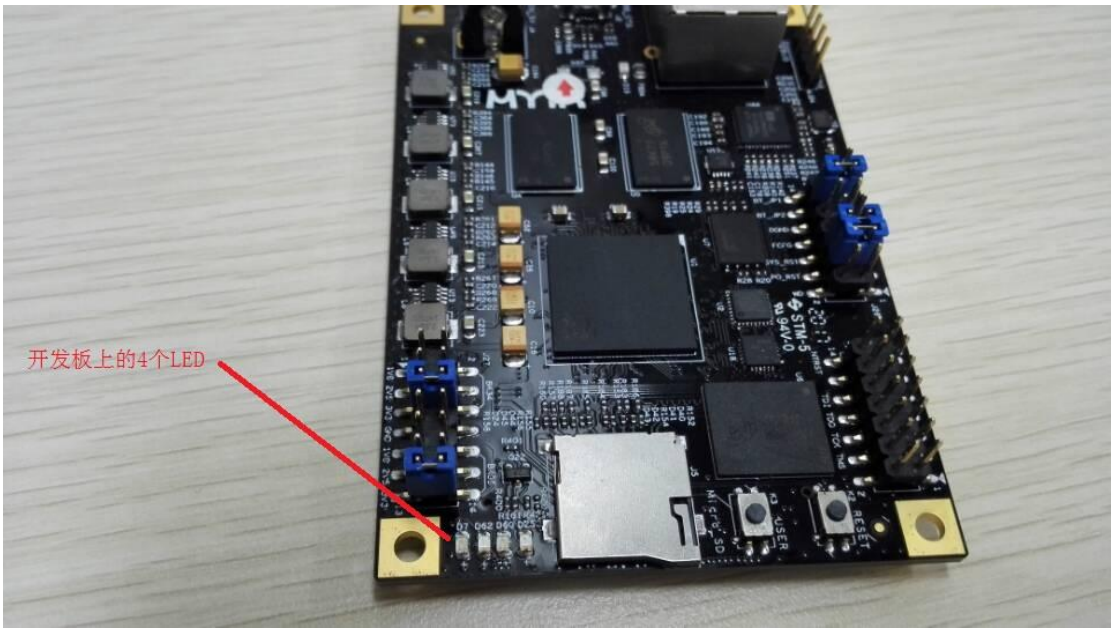
1. 将 JP26 的 BT_JP2 跳线闭合，jp1 断开。
2. 将 xilinx usb 下载器 DLC9 连接到开发板的 J1 JTAG 端口和电脑的 USB 端口，这样就 PC 就通过 JTAG 接口连接到开发板了
3. 将 MY-UART012U USB 串口线连接到开发板的 J24 UART 端口和电脑的 USB 端口，这样 PC 就通过串口连接到开发板上了
4. 将我们提供的 5V 电源连接到 z-turn_Lite 的 J4
5. 将以太网的网线连接到 z-turn_Lite 的千兆以太网端口的 J3 RJ45 连接器，并将另一端连接到电脑的网线接口上
6. 启动 Putty 调试工具，将串口的波特率设置为 115200、8 位、1 位停止位，无奇偶校验和无流量控制（请参阅本文档末尾的设置主机 PC 部分，以安装 用于 USB-UART 端口的软件驱动程序，并设置 UART）。
7. 将电脑的 IP 地址设置为 192.168.1.1，子网掩码为 255.255.255.0

用户接口

编号	说明	备注
J1	JTAG	JTAG 下载器插口
J2	USB-OTG	开发板上的 USB-OTG 接口
J3	10/100/1000 ETH	RJ45 千兆以太网接口
J4	5V DC IN	开发板 5V 直流电源接口
J5	TF-Card	开发板上的 TF 卡卡槽
J24	Debug UART	开发板上的 UART
J26	Boot Select	启动模式选择
J27	Bank Power Select	可以切换 Bank 的工作电压
K2	Reset Button	开发板的复位按钮
K3	User Button	可供用户使用的按钮
U1	xc7z007s/xc7z010	ZYNQ 芯片
U4	512M DDR	开发板上的 DDR
U5	512M DDR	开发板上的 DDR
U6	4GB emmc	4GB eMMC
U7	16MB QSPI_Flash	16M QSPI_Flash

LED 说明

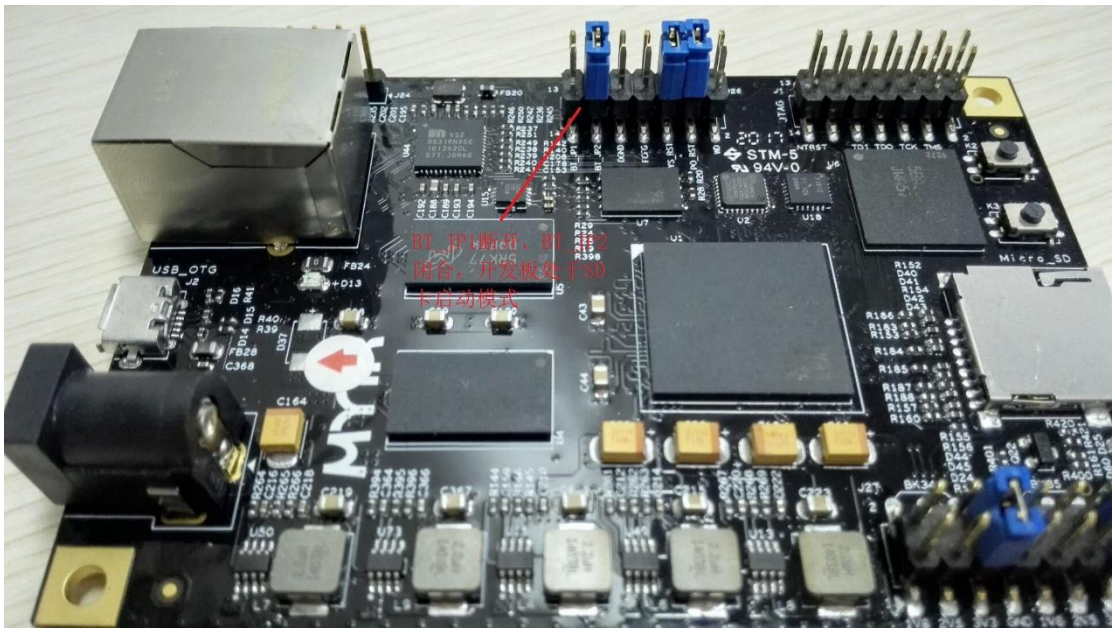
编号	说明	备注
D25	Power indicator	开发板上的电源指示灯
D60	PS_User_LED	接到 PS 可供用户使用的 LED
D62	FPGA Init error	FPGA 初始化失败
D7	FPGA_Config_Done	开发板烧写成功指示灯



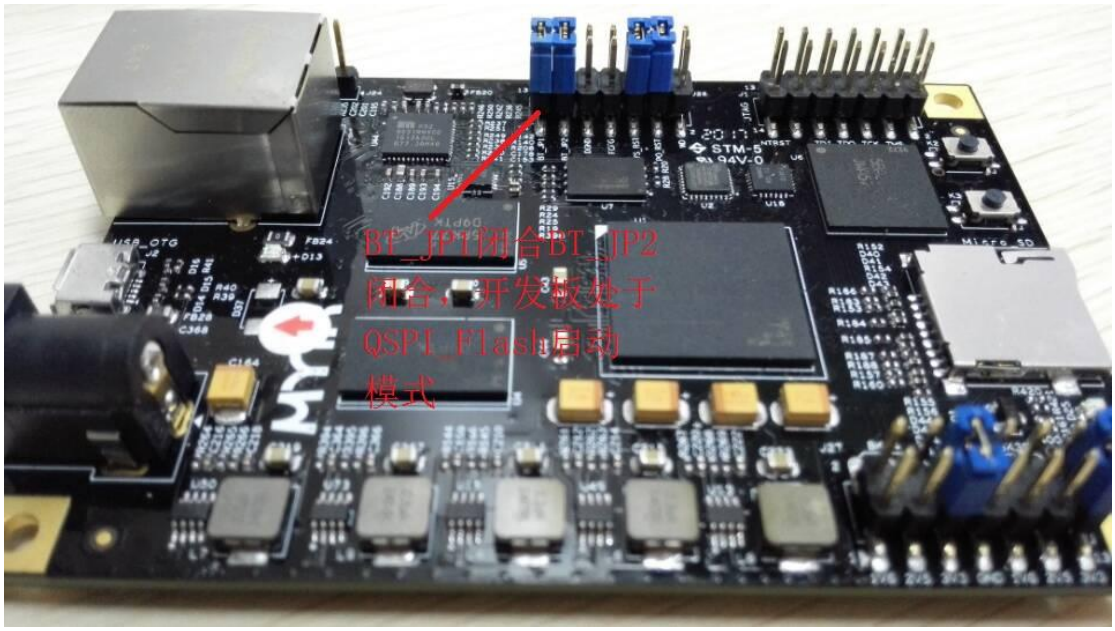
开发板模式介绍

编号	BT_JP1	BT_JP2	备注
J26	OFF	ON	SD 卡启动模式
J26	ON	ON	QSPI_Flash 启动模式

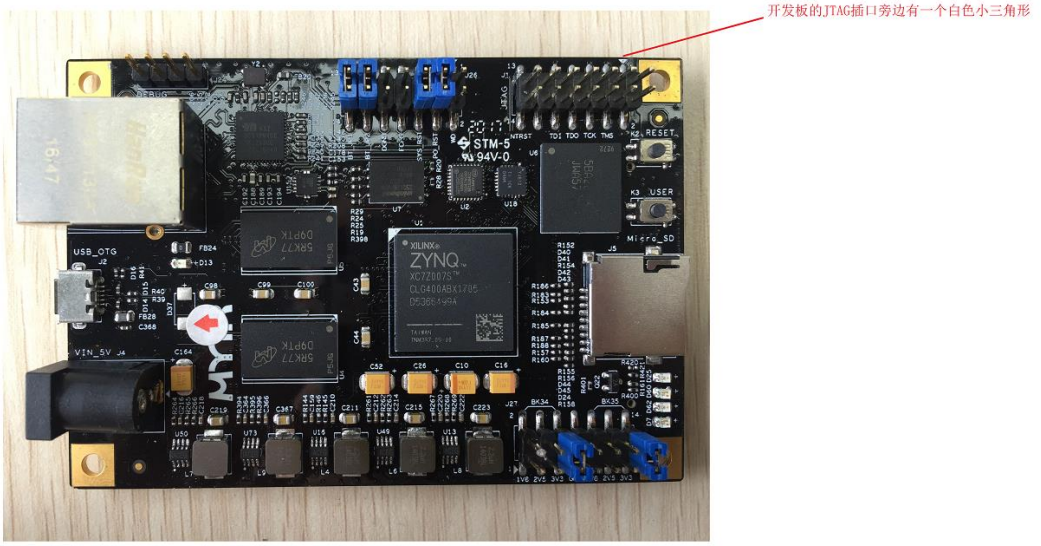
SD 卡启动模式



QSPI_Flash 启动模式

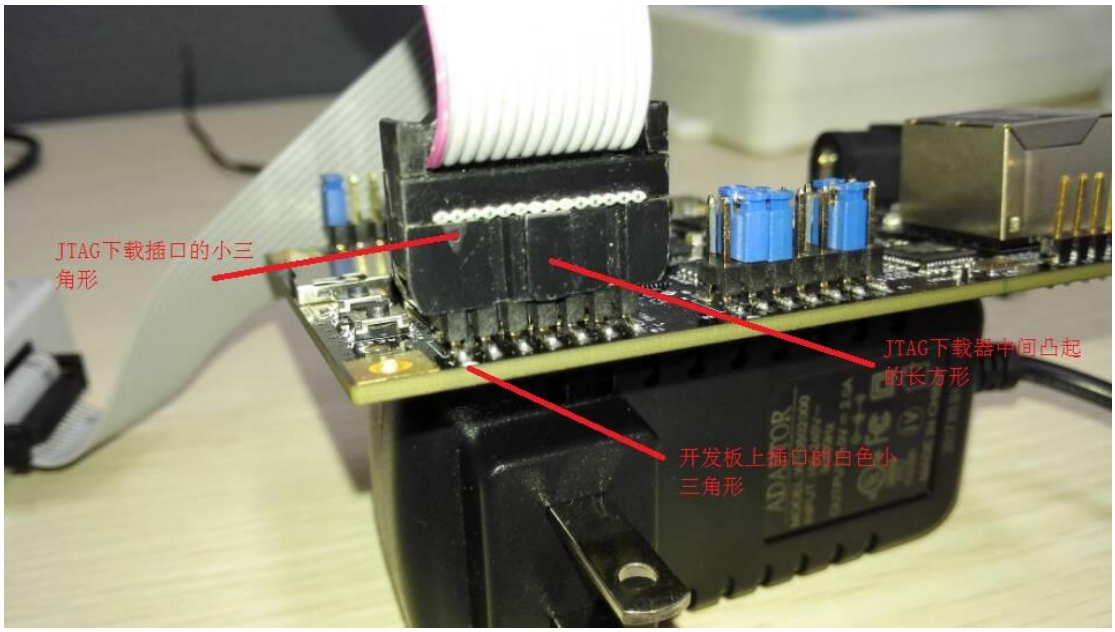


JTAG 接法

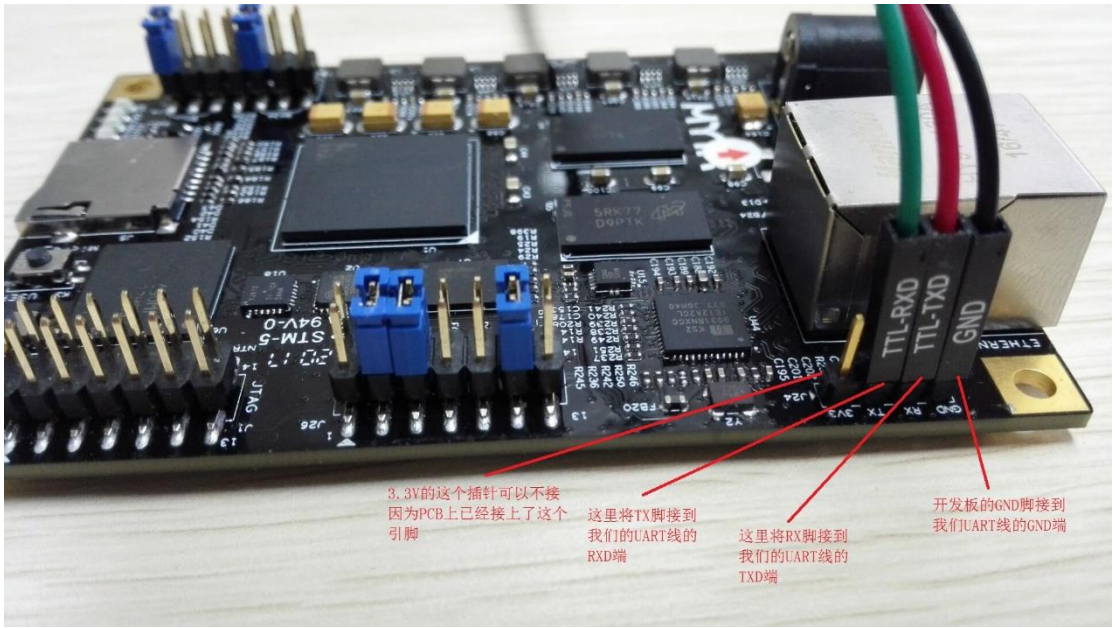

开发板的JTAG插口旁边有一个白色小三角形


JTAG下载器的插口处有一个小三角形
JTAG下载器的插口中间有一个凸起的长方形

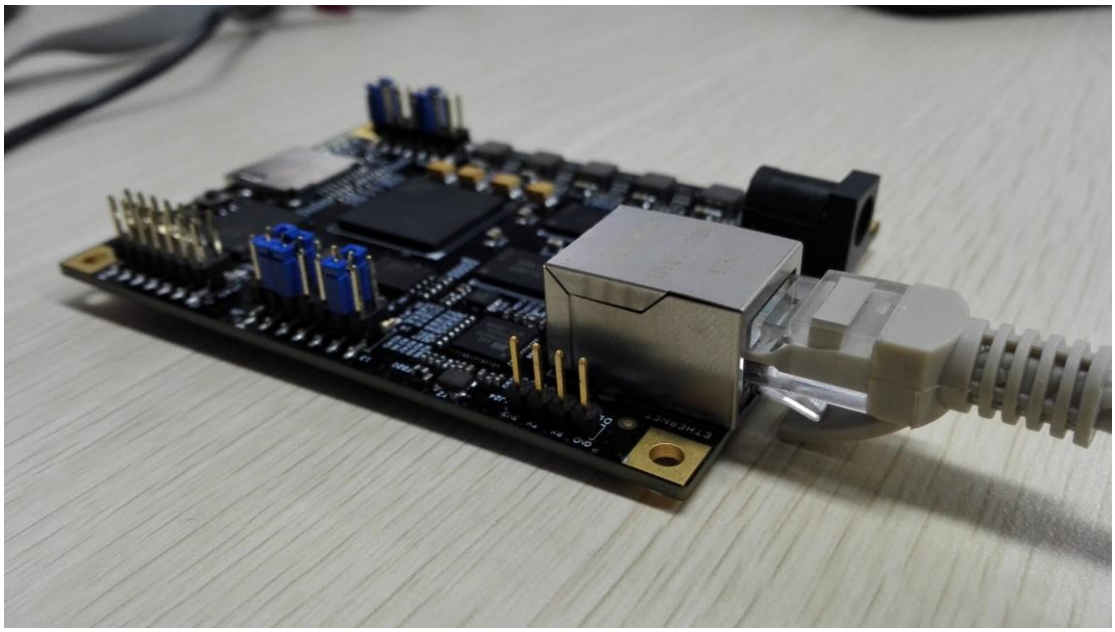
将这个三个都在同一面，然后将下载器的 JTAG 插口插到开发板的 JTAG 插针上



USB_UART 接法

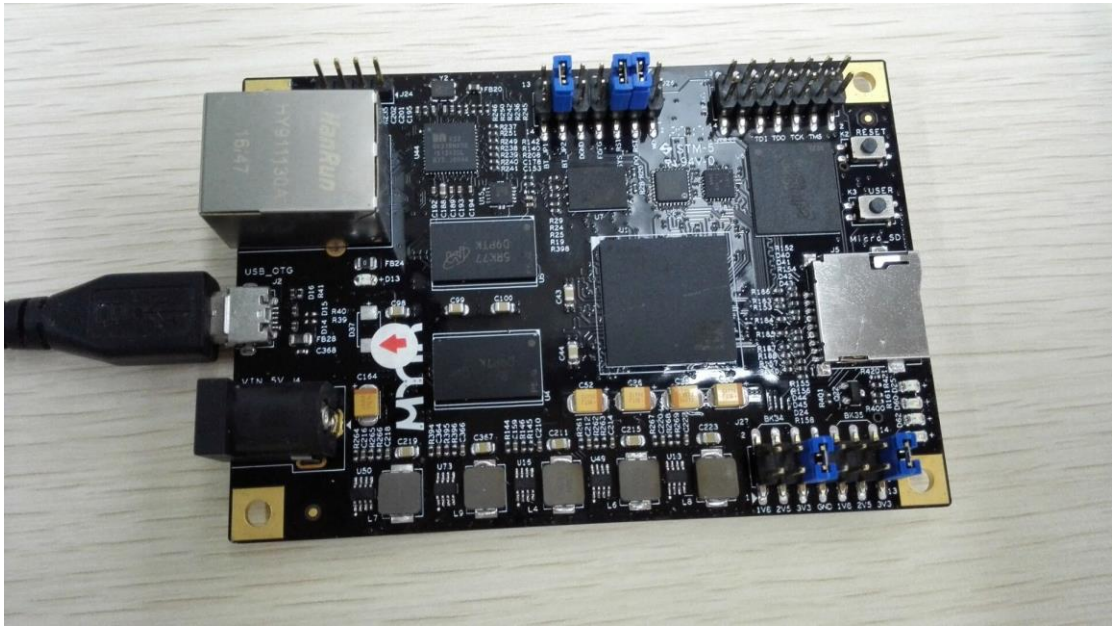


以太网只要将 RJ45 水晶头对准以太网口插入就可以了

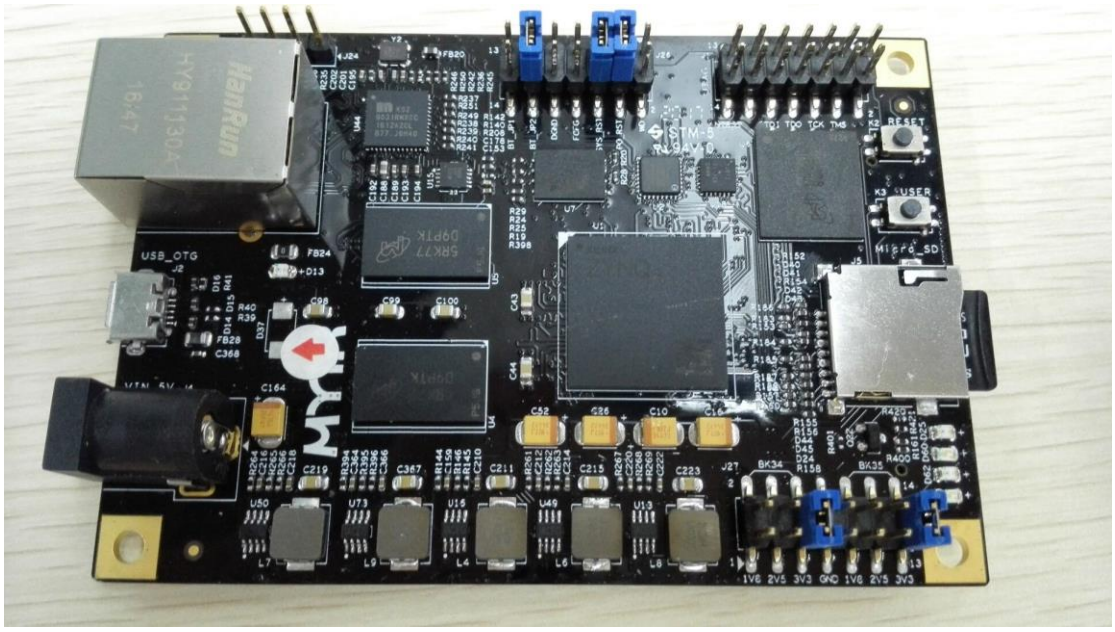


5v 直流电源的接法

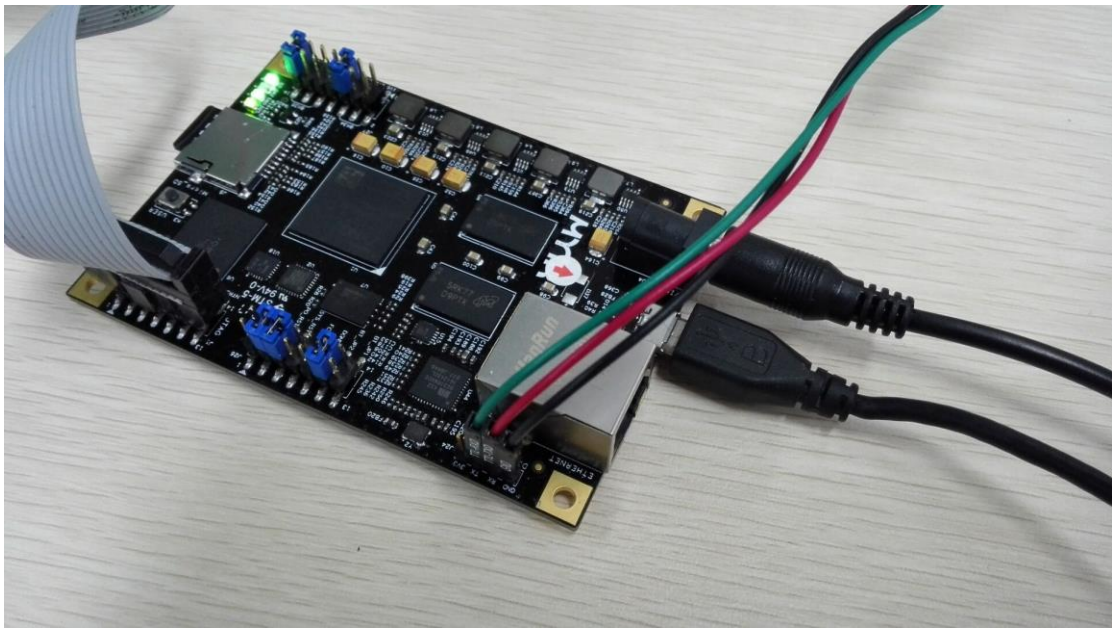
USB-OTG 接法



TF-Card 接法



开发板接上所有的设备如下图所示

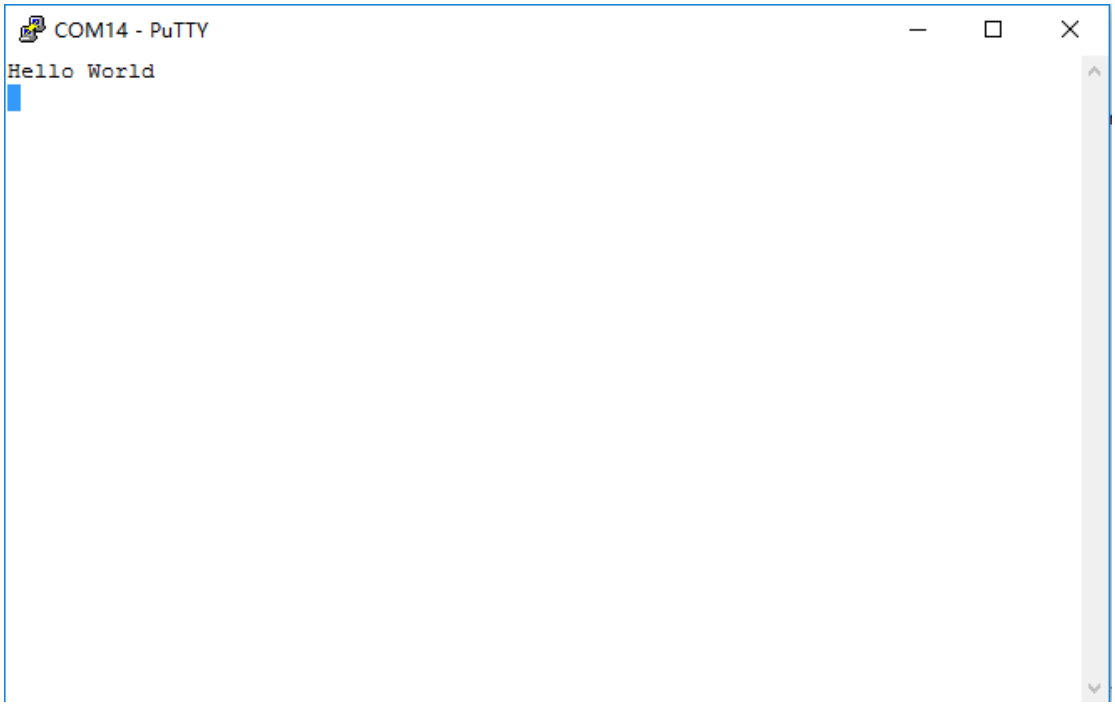


7 在硬件上运行 Hello World 程序

在本节中，我们将使用 JTAG 调试接口将可执行文件 hello_myir.elf 加载到 Z-Turn-Lite DDR3 内存并运行。

•从 SDK GUI Project Explorer 中，右键单击 hello_myir 软件项目，然后选择 Run As> 1 Launch on Hardware（系统调试器），如下图所示。 该命令将将 hello_myir.elf 可执行文件加载到 Z-Turn-Lite DDR3 内存中，并在板上运行。

你可以看到 putty 上面打印出 Hello World



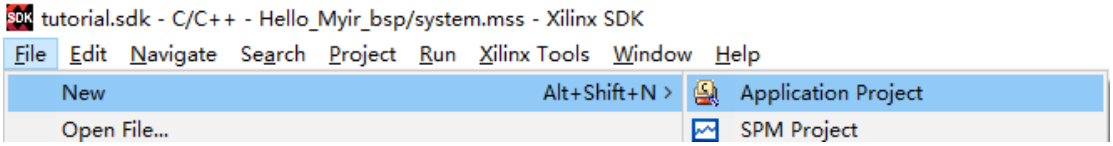
8 建立应用程序

现在,一个简单的 Hello World 程序已经被加载到 DDR3 运行,在接下来的部分将尝试将其他应用程序加载到 DDR3 中，并在开发板上运行。

8.1 让 PS User LED 闪烁

本节我们将做一个简单的工程，对开发板上 PS 部分的用户可使用的 LED 进行操作，第一步先在 SDK 中创建一个 PS_User_LED 的工程。

·在 SDK 菜单栏上单击 **File > New > Application Project**，如下图所示

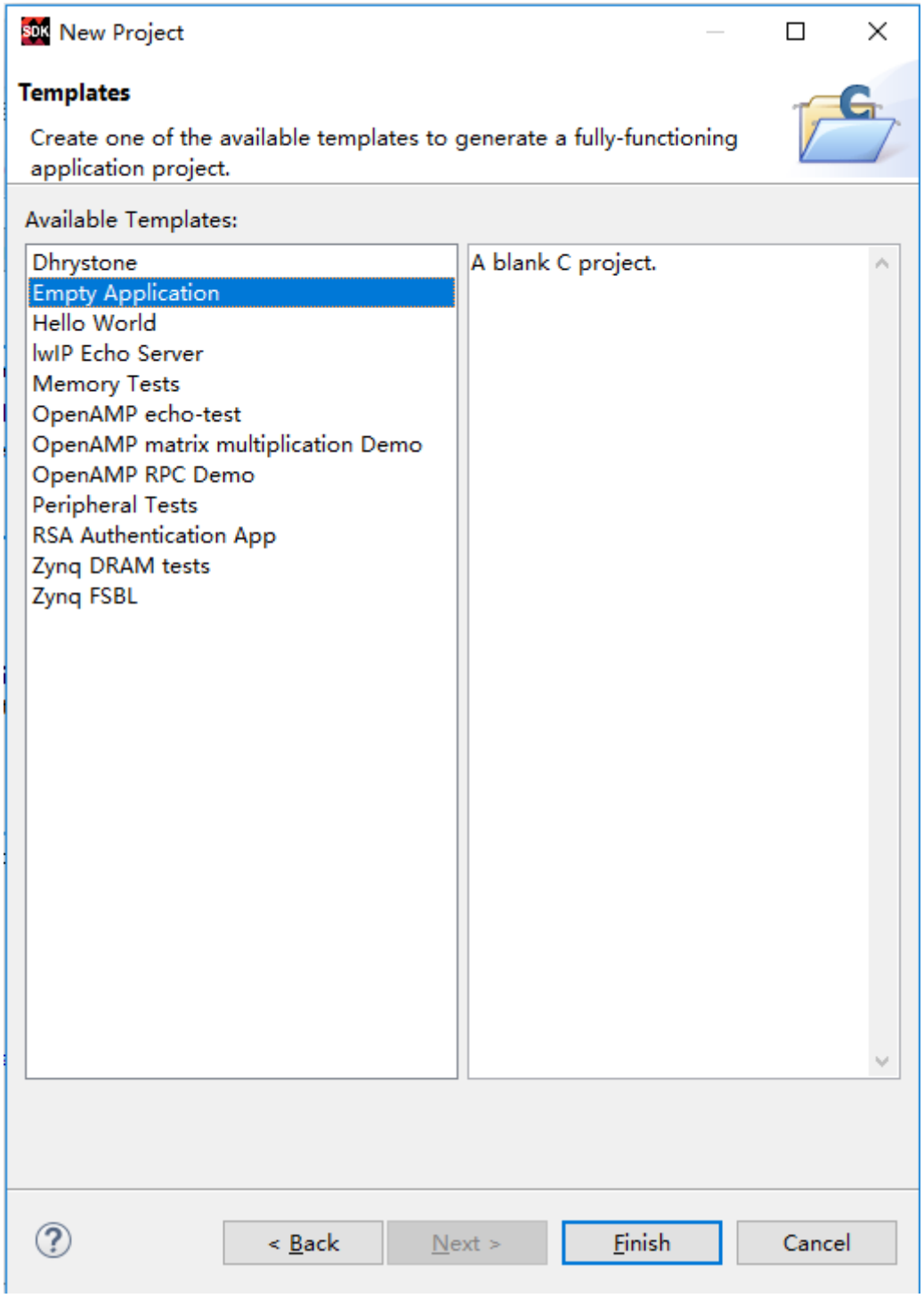


当弹出新建工程对话框时：
a 在弹出的对话框中设置工程名为 PS_User_LED
b 然后在 Board Support Package 中单击 Use existing 单选按钮选择生成 Hello_Myir_bsp
c 其它的设置默认保持不变然后单击 Next

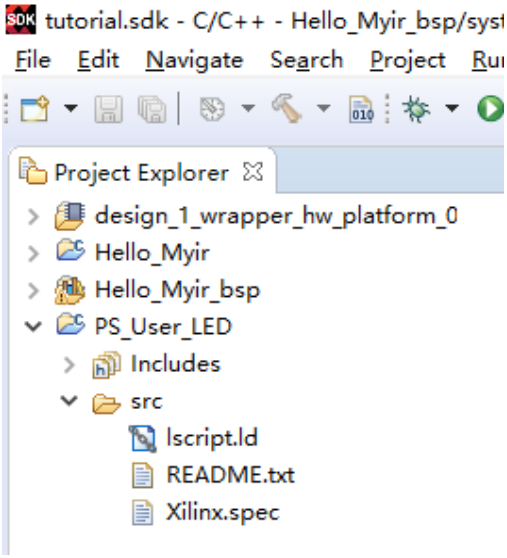
在弹出的对话框中：

a 我们选择创建一个空的 **PS_User_LED** 工程文件，这个空的工程没有添加主程序，所以下一步我们将介绍怎样添加一个主程序

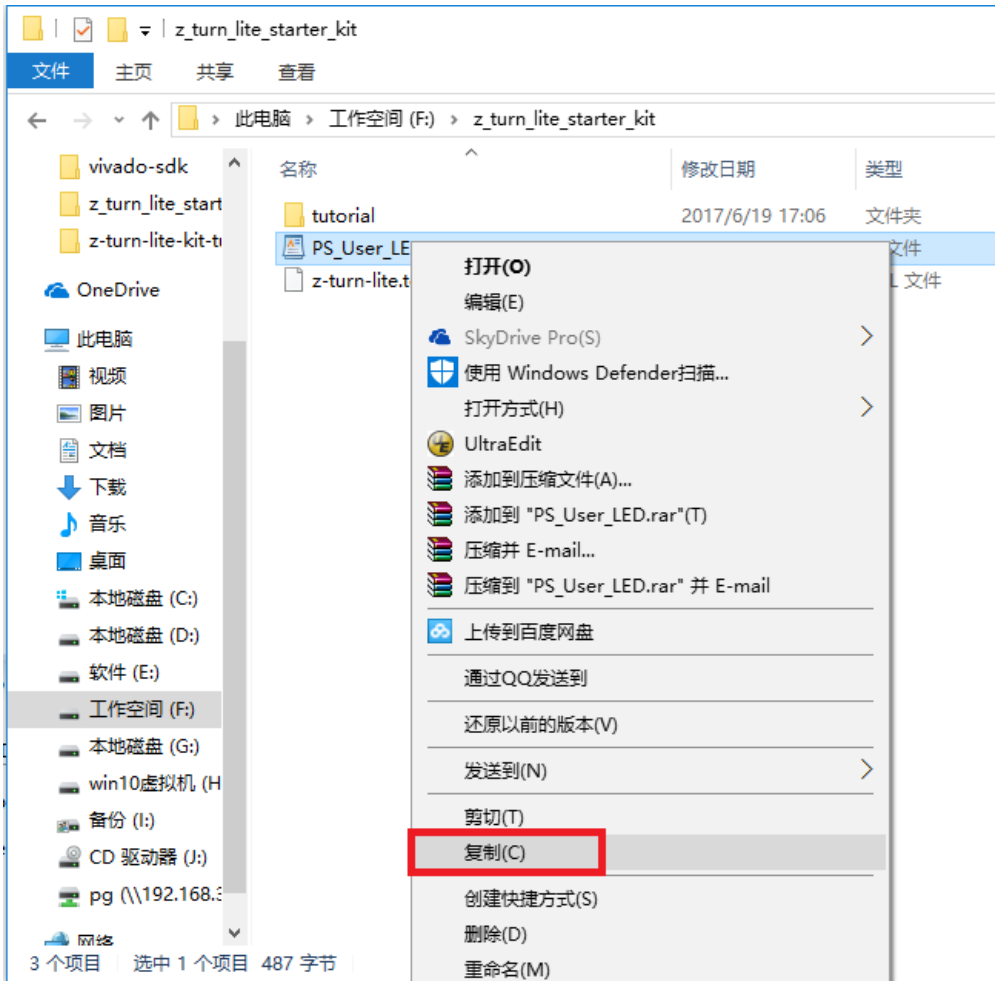
b 然后点击 Finish



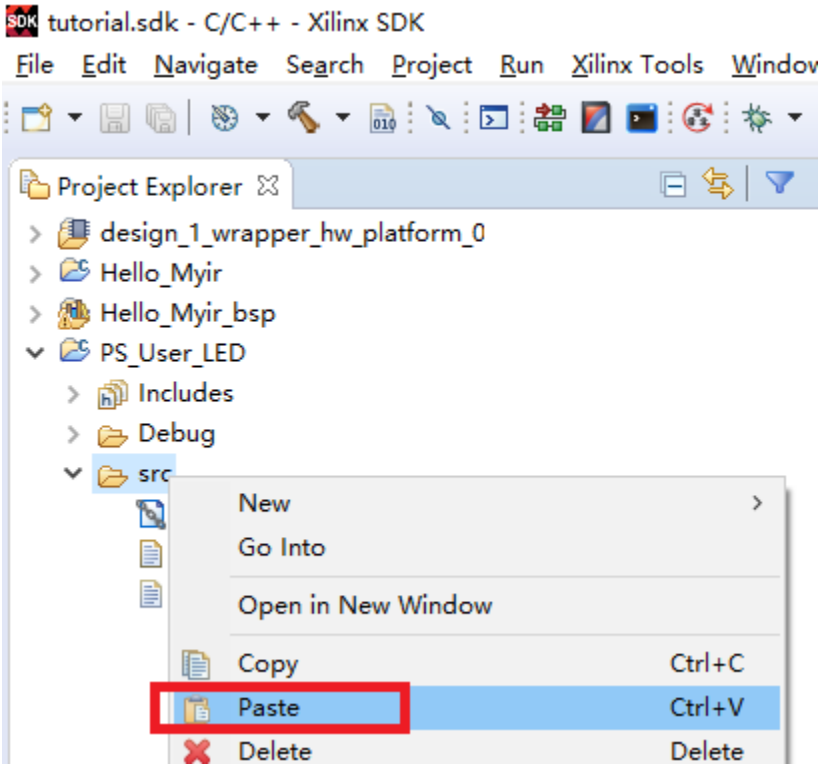
在 SDK 工程管理器下，展开 PS_User_LED 下的 src 文件夹，你可以看到没有我们可以编辑的主程序，我们将在下一步骤中演示怎么添加主程序。

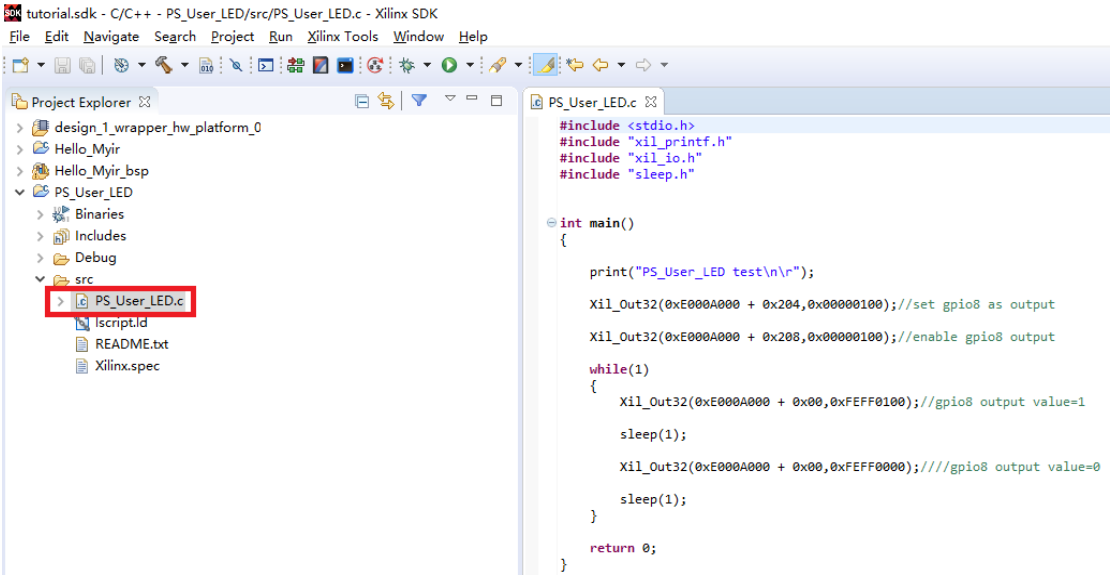


复制 **F:/z_turn_lite_starter_kit** 路径下的 **PS_User_LED.c** 文件

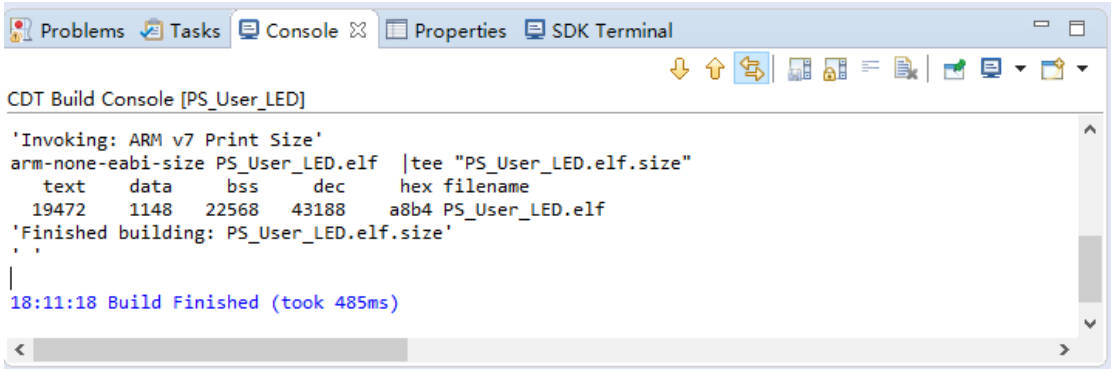


将 **PS_User_LED.c** 这个文件粘贴到 src 目录下，如下图示所示

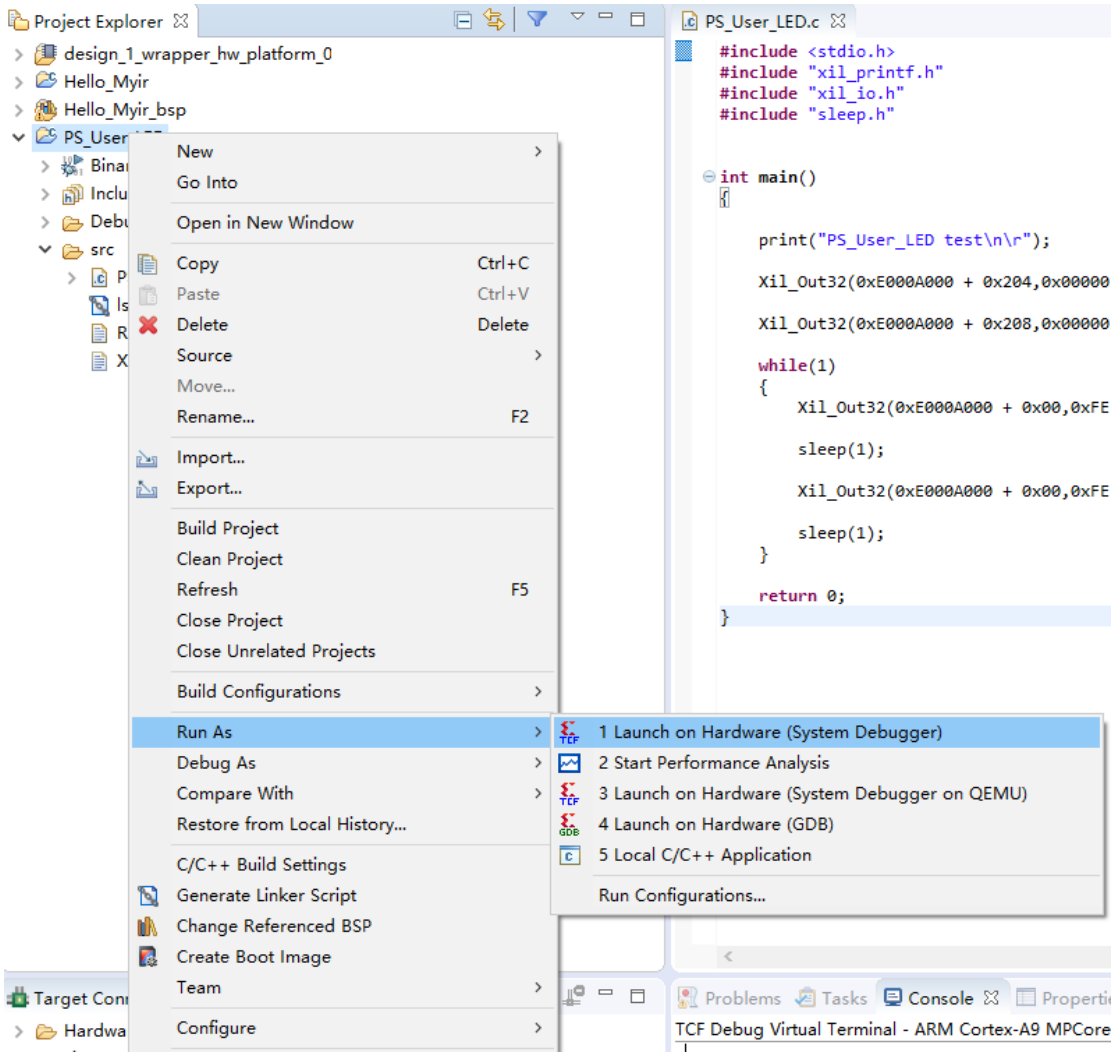




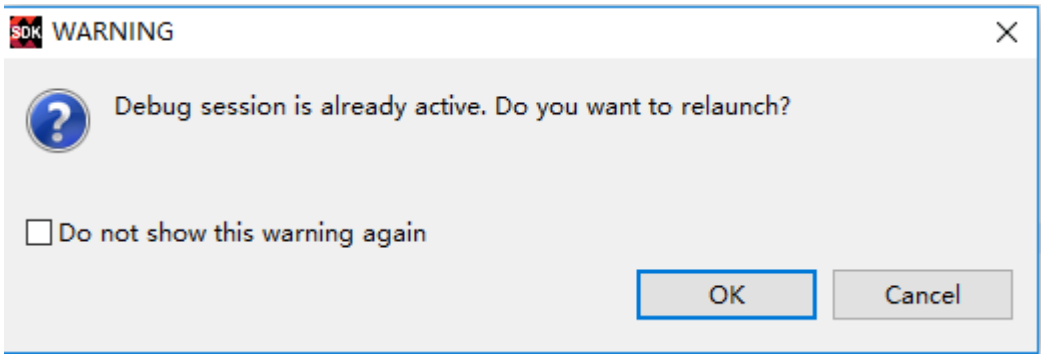
在 SDK 窗口中可以看到正在编译的 PS_User_LED 项目，如下图所示。 生成的 PS_User_LED.elf 将被加载到 Z-Turn-Lite DDR3 内存中并在下一步中执行的编译器的输出。



•在这个 SDK 的工程管理器下，选中 **PS_User_LED** 文件夹右击选择 **Run As > 1 Launch on Hardware (System Debugger)** ， **PS_User_LED.elf** 可执行文件将被加载到 DDR3 内存并在板上运行，如下图所示。



•在弹出的对话框中单击 OK

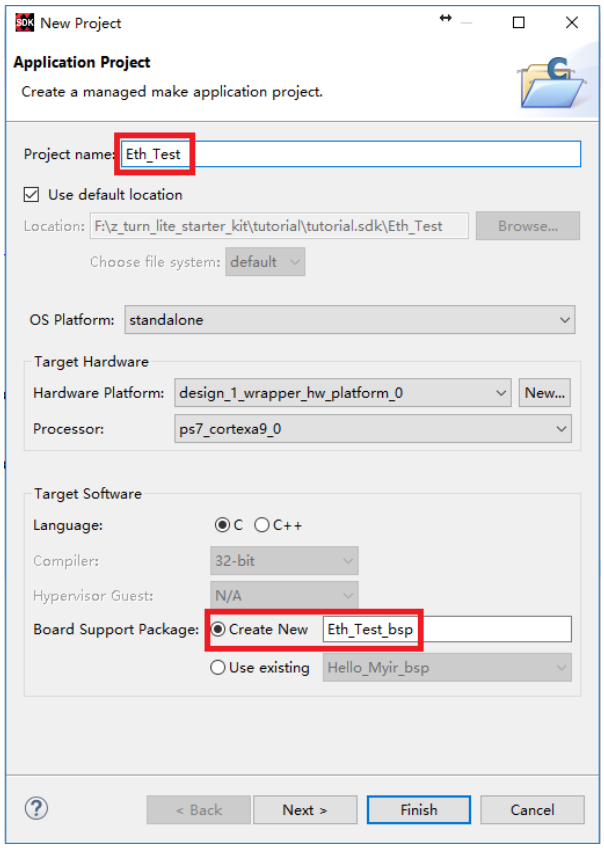


你可以看到开发板上由用户控制的 PS 部分的 LED 灯 D60 不停的闪烁。

8.2 千兆以太网端口测试

在本节中,我们将创建一个 SDK 工程来验证 Z-Turn-Lite 开发板上的千兆以太网端口。

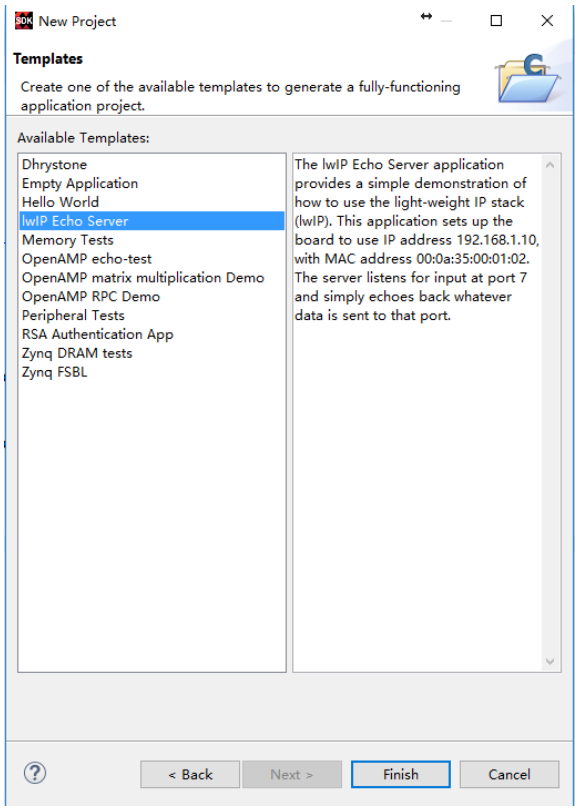
•点击菜单栏上的 File > New > Application Project.
•在弹出的对话框中：
a 在弹出的对话框中工程名为 Eth_Test
b 然后在 Board Support Package 中单击 Create New 按钮生成 Eth_Test_bsp
c 其它的设置默认保持不变然后单击 Next



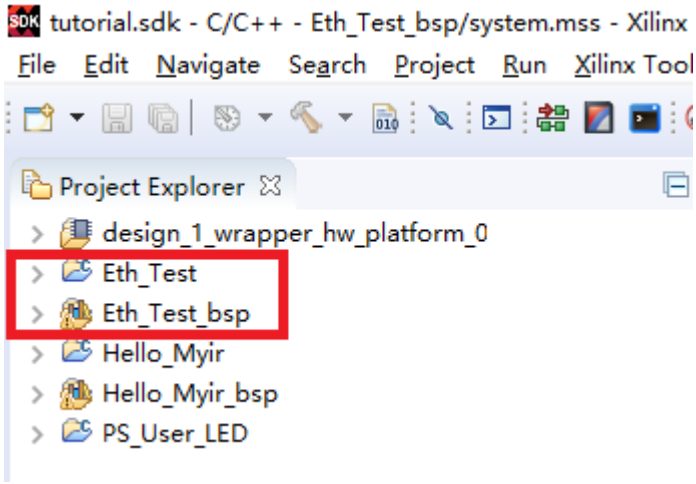
在弹出的对话框中:

a 选择 **lwIP Echo Server**

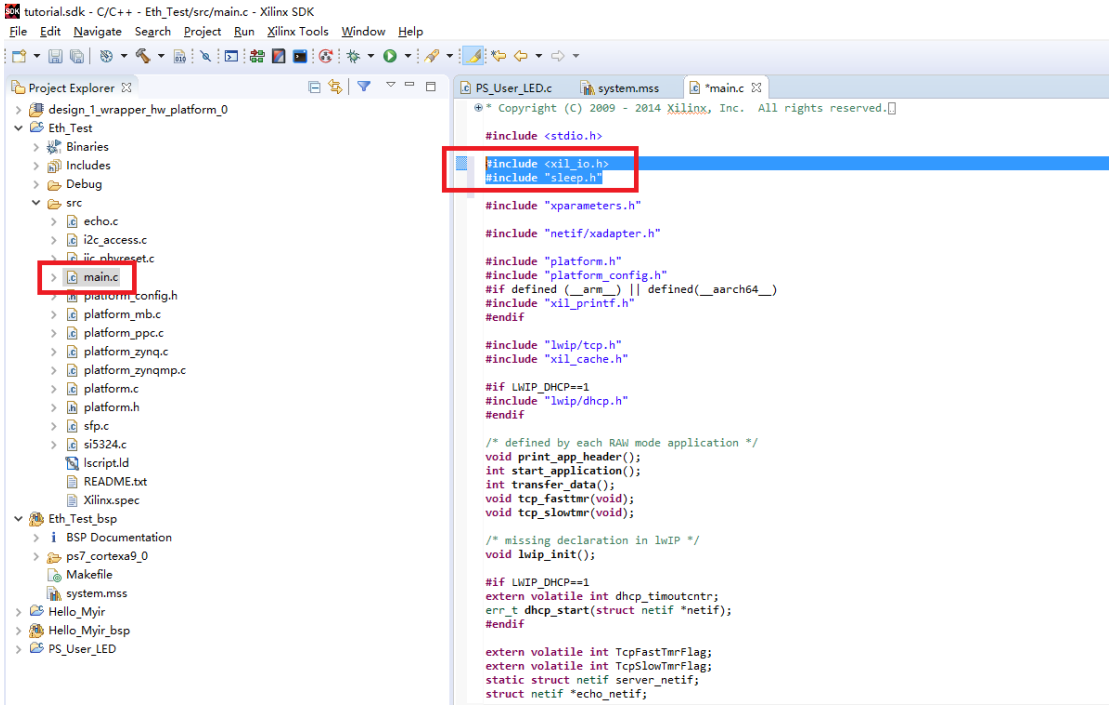
b 然后单击 **Finsh**

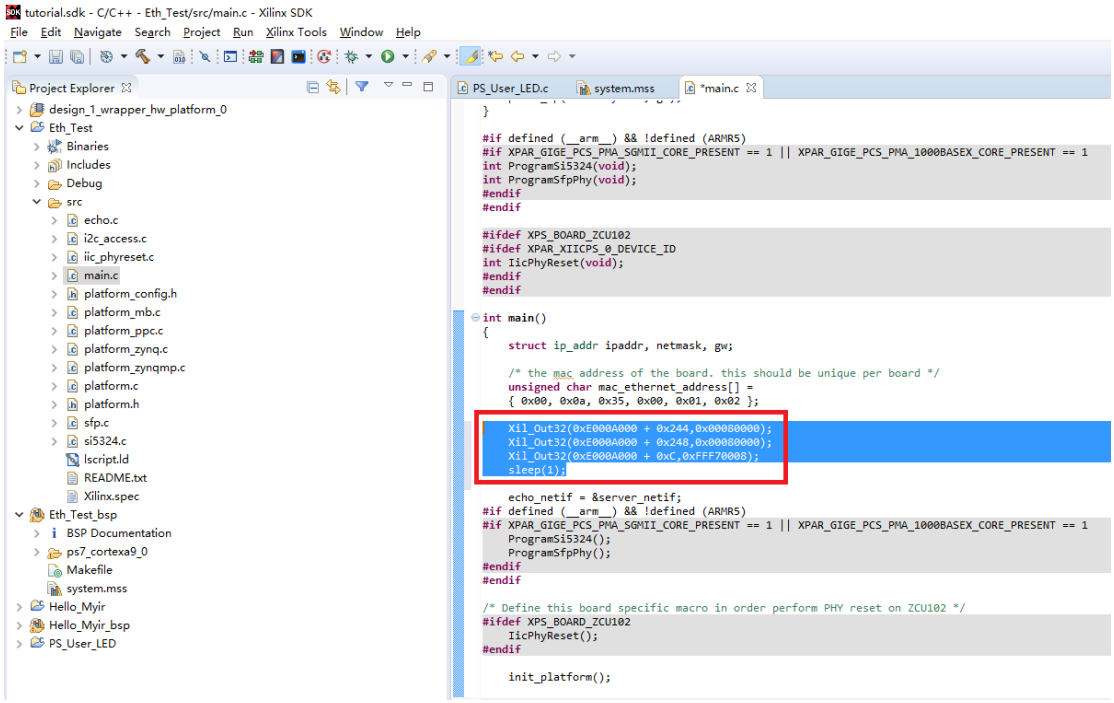


我们将看到生成一个 **Eth_Test** 工程和 **Eth_Test_bsp** 的 **BSP** 文件



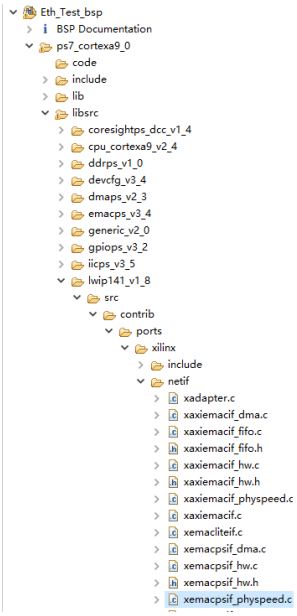
将我们提供的工程里的 main.c 文件用红框标注的地方复制到你自己新建的工程的相同位置，然后按 ctrl+s 保存。



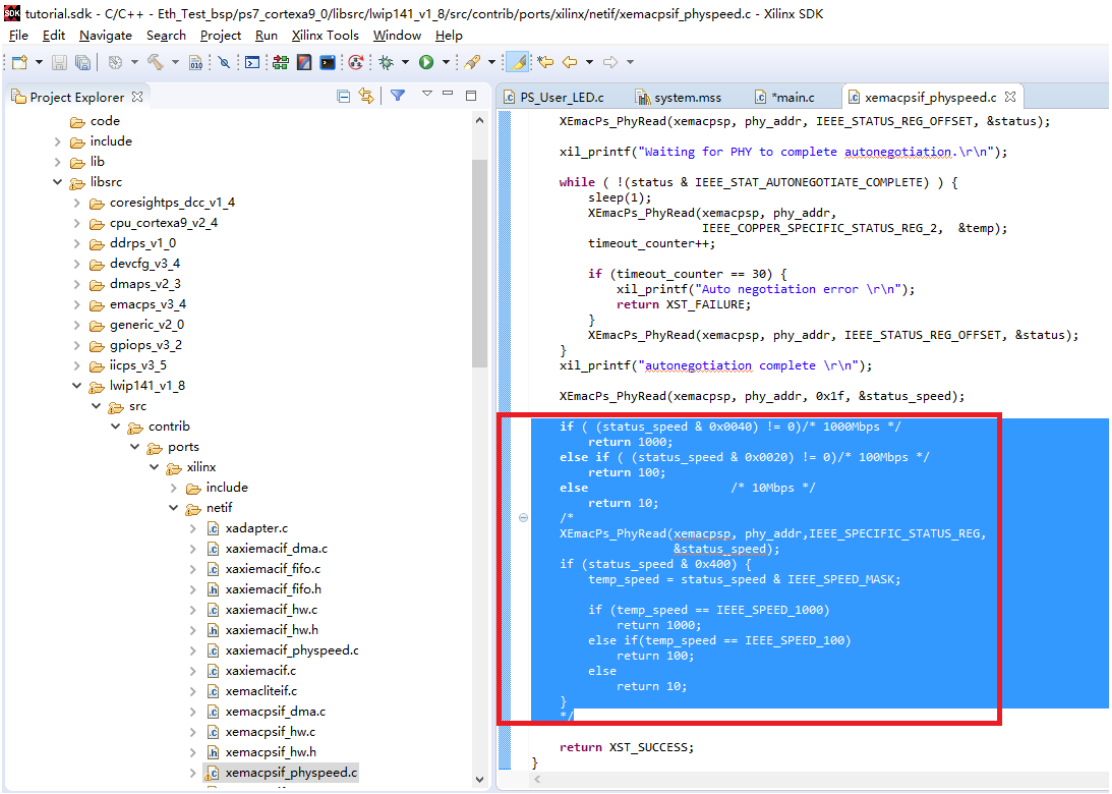


打开路径下：

Eth_Test_bsp->ps7_cortexa9_0->libsrc\lwip141_v1_8->src->contrib->ports->xilinx->netif-> xemacpsif_physpeed.c 文件按 Ctrl+s 保存，并且会自动编译生 Eth_Test.elf 文件，在下一节中，我们将将 Eth_Test.elf 可执行文件加载到 DDR3 内存中，并在板上运行。

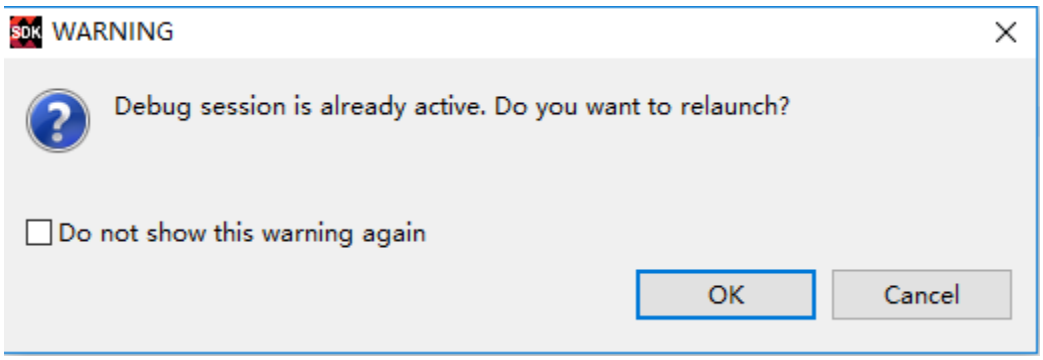


在这个函数末尾 **static u32_t get_Marvell_phy_speed(XEmacPs *xemacpsp, u32_t phy_addr)**将我们提供的工程相同地方的文件覆盖你自己新建工程的相同地方的文件。

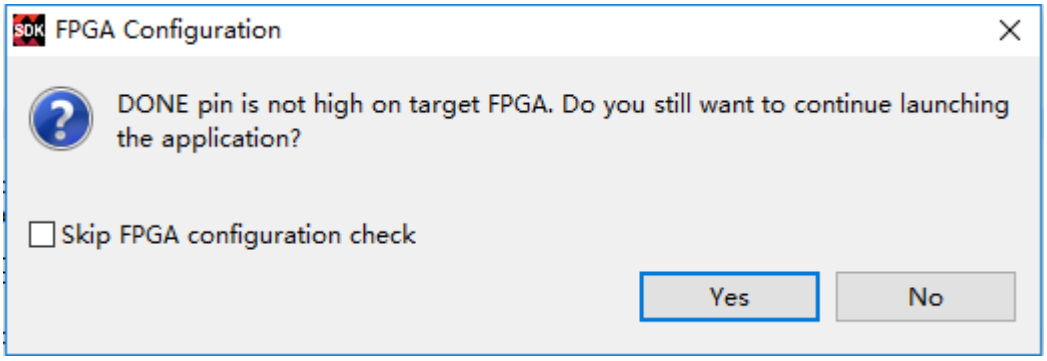


•在 SDK 工程管理器下，单击选中 **Eth_Test 然后右击选择 Run As > 1 Launch on Hardware (System Debugger)**，生成的 Eth_Test.elf 可执行文件将被加载到 DDR3 内存中，并在板上运行。

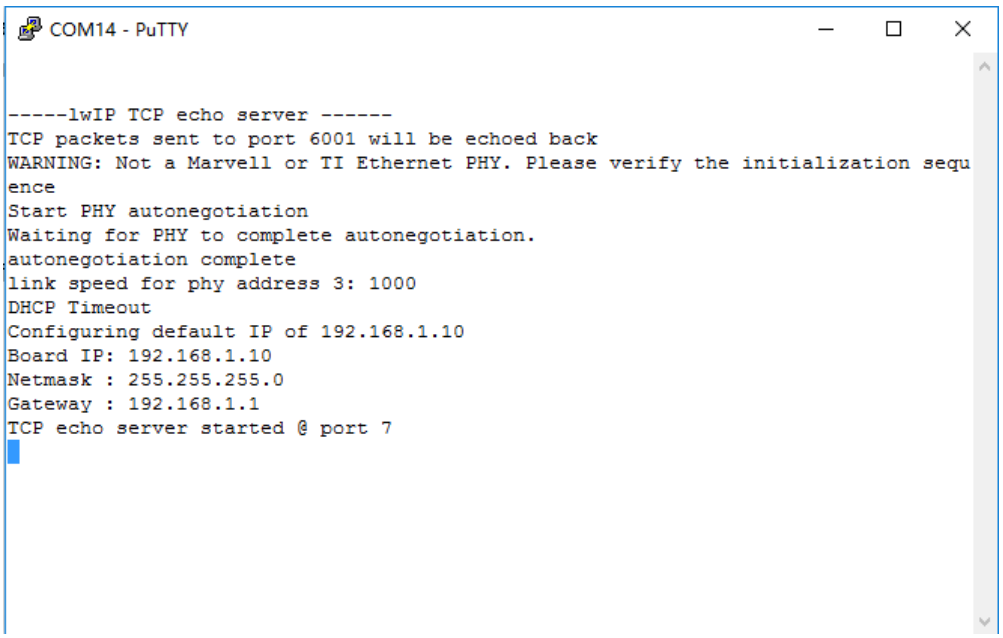
•在出现的对话框中，单击 OK



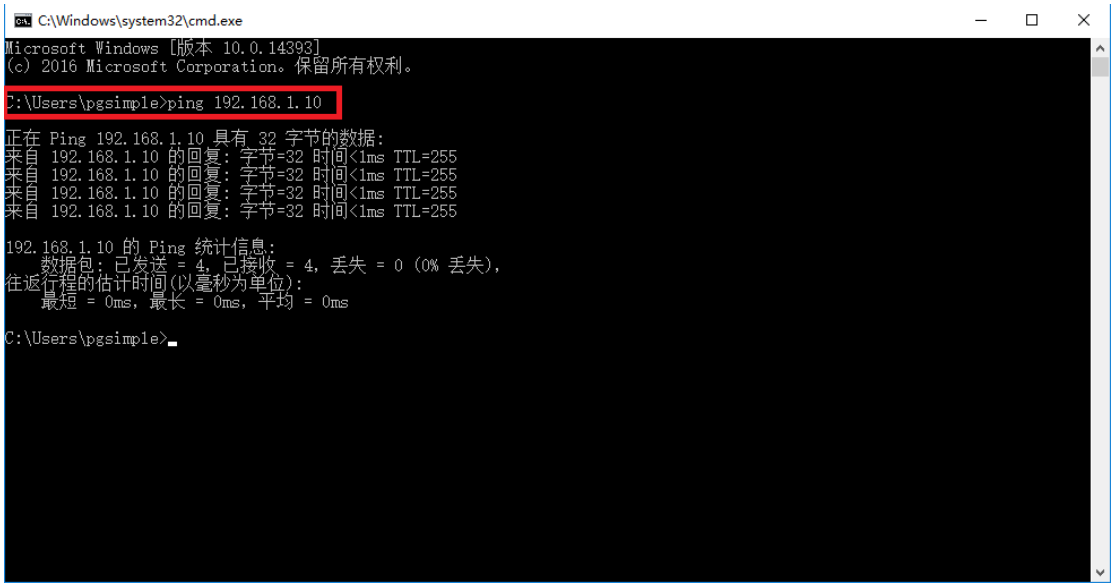
•在继续弹出的对话框中单击 Yes



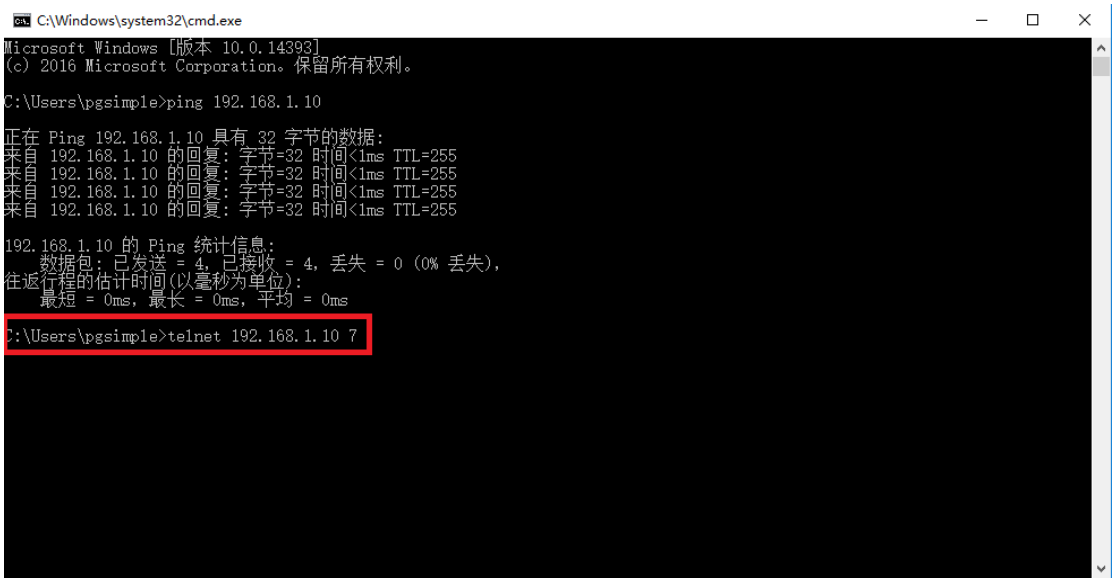
•可以从 Putty 上面看到 lwIP Echo Server 工程已经自动的分配了一个 IP 地址，说明 lwIP Echo Server 工程可以正常运行。



•打开我们 window 操作系统的 CMD 调试窗口，去 ping 这个自动分配的 IP 地址 **192.168.1.10** 可以看到回复了 4 个 32 位，说明 lwIP Echo Server 已经正常运行。



•要连接 echo 服务器，请使用 telnet 实用程序。 键入以下 telnet 命令，如下所示，然后按返回键。



如果 echo 服务器正常工作，那么发送过去的数据都会有响应，telnet 客户端会立即将字符发送到服务器上并将接受的数据回传到调试端口，你只需发送几个字符就可以看到他们在终端上回传。

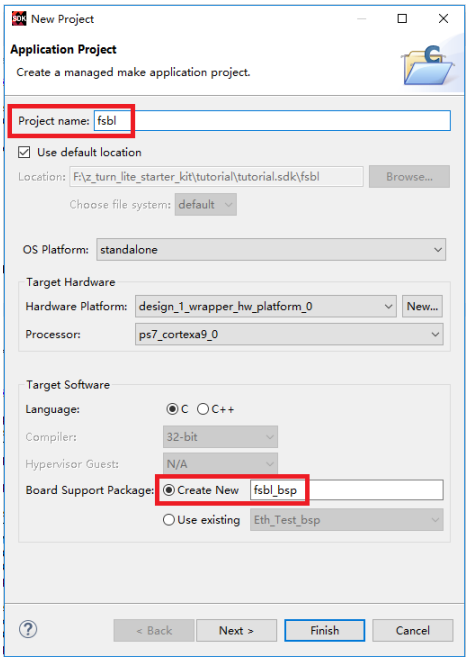
9 从主/辅助设备启动

在前面的章节中，SDK 调试器用于加载用户的可执行文件到 z_turn_lite DDR3 内存在 JTAG 接口板上运行的程序。这一章我们介绍如何通过 sd 卡和 qspi 来启动开发版，并将应用程序加载到 z_turn_lite DDR3 运行。

接下里的部分会指导你：
•生成各个启动设备的启动镜像
•烧写镜像到启动设备
•从各个启动设备启动

9.1 产生 Boot Loader（fsbl）

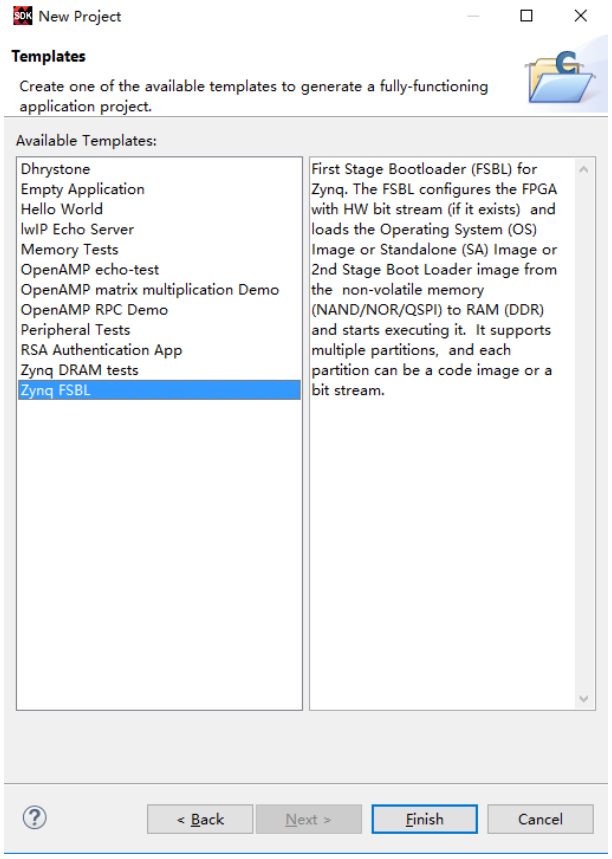
单击 SDK 菜单栏选择 **File > New > Application Project**
在出现的新建工程对话中：
a.填写工程名为 **fsbl**
b.在 **Board Support Package** 栏选择 **Create New** 新建一个 **fsbl_bsp** 文件
c.保持所有其他默认选项，然后单击下一步



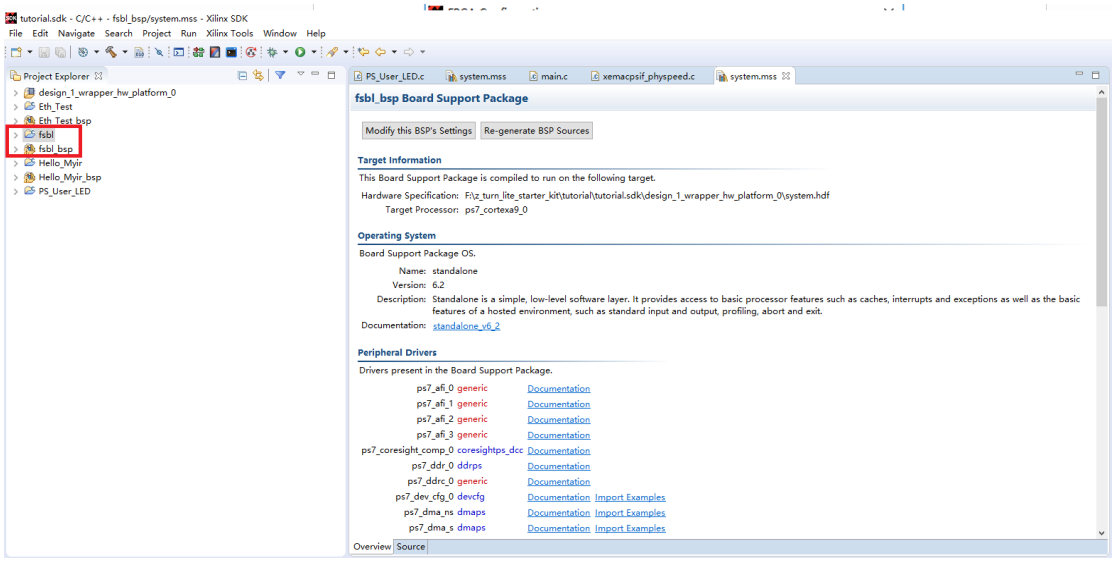
在弹出的对话框中:

a. 选择 **zynq FSBL**

b. 然后单击 **Finish**



然后可以看到 **SDK** 中生成了 **fsbl** 工程和 **fsbl.bsp** 文件，并且将自动编译并且生成可执行文件 fsbl.elf 文件，如下图所示。

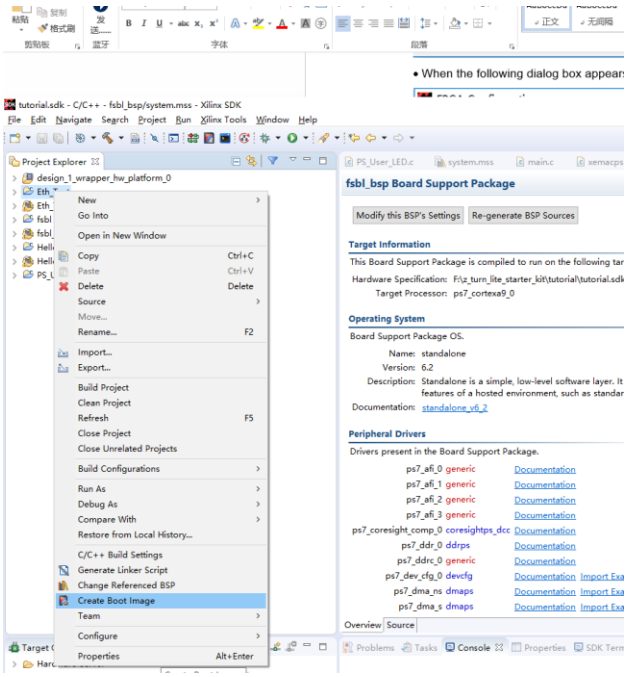


9.2 从 microSD 卡引导

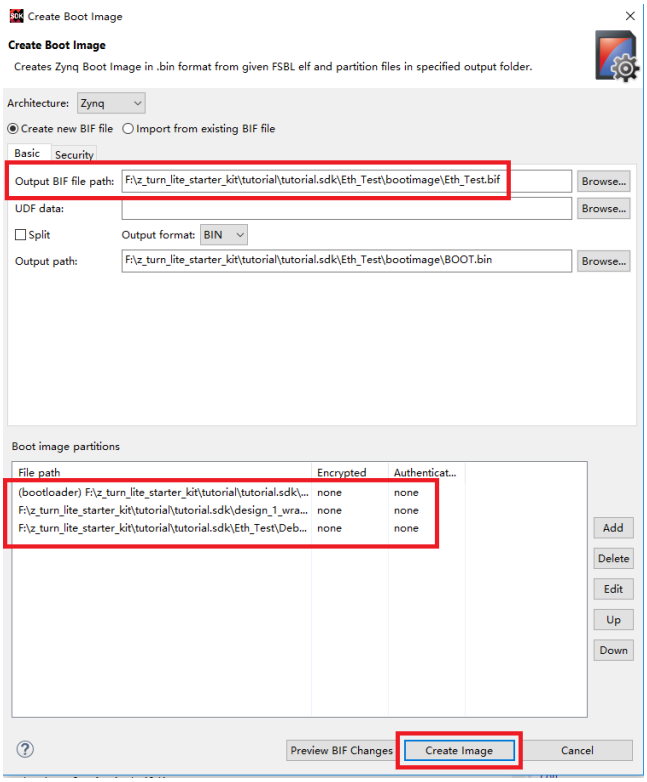
为了从 microSD 卡启动，生成引导映像并存储在 microSD 卡上。

9.2.1 生成 SD 卡启动映像

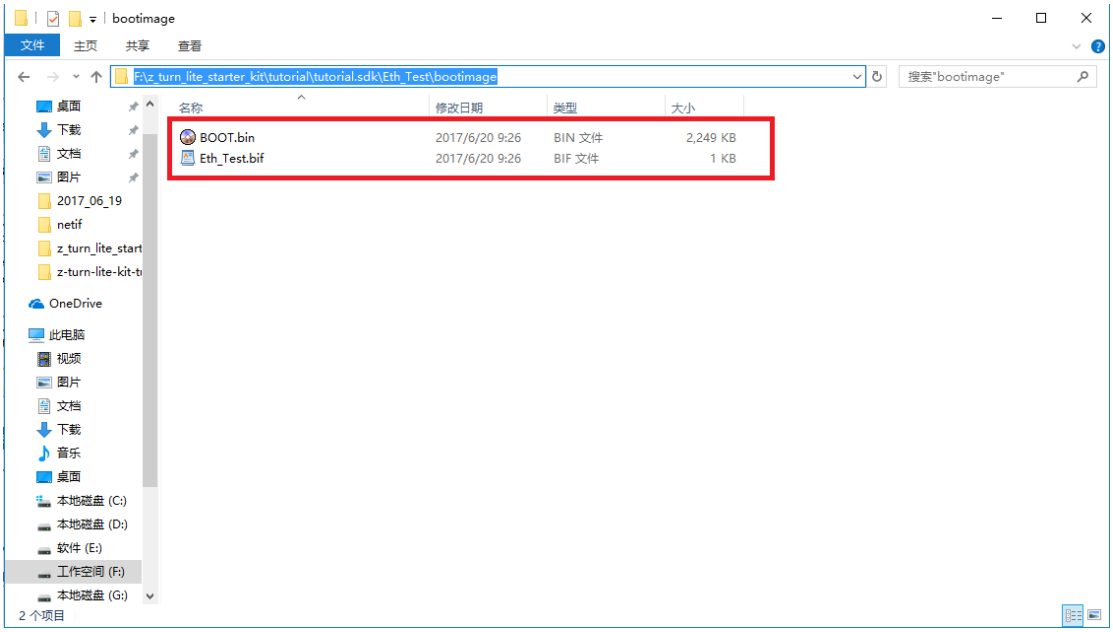
•从 SDK 工程管理器中，右键单击 Eth_Test 选择 **Create Boot Image**，如下图所示



单击 **Create Image**



在 **Eth_test/bootimage** 文件夹下会生成 **BOOT.bin** 和 **Eth_Test.bif** 两个文件



注意：BOOT.bin 文件已经为您生成并存储在 ready_to_download 和 bin_and_mcs_generation_solution 文件夹中，以防您生成此引导映像文件时遇到其它的问题。

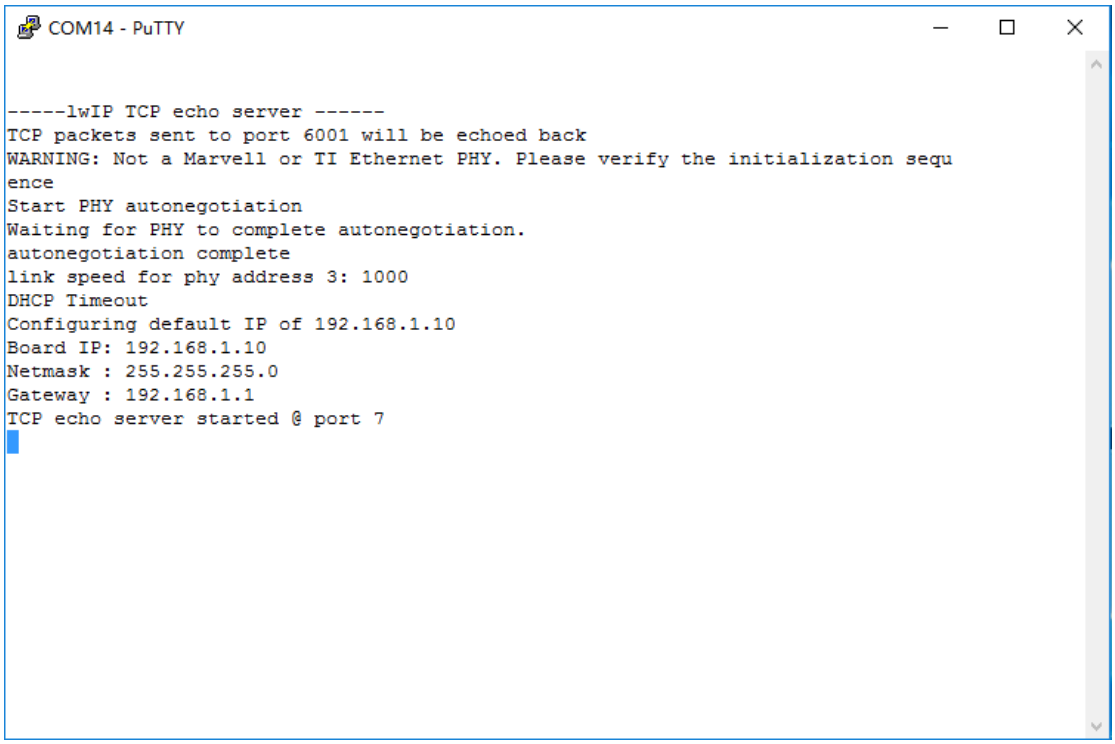
9.2.2 将 SD 卡启动映像复制到 microSD 卡上

米尔 Z-Turn-Lite 入门套件随附 4GB microSD 卡。 使用 PC 的存储卡读卡器插槽，将上一步中生成的 BOOT.bin 文件复制到 microSD 卡的根目录下（您也可以使用 USB 存储卡读卡器适配器，如果您的电脑没有 microSD 卡读卡器插槽）。 从读卡器中取出 microSD 卡。

注意：在执行上述 BOOT.bin 复制任务之前，请确保将我们提供的 microSD 卡的原始内容保存在 PC 硬盘驱动器上。

9.2.3 使用 microSD 卡启动开发板

•在 JP26 的 BT_JP2 上安装跳线帽，并确保 JP26 的 BT_JP1 上没有跳线帽。
•在 Z-Turn-Lite 开发板上将 5V 电源连接到 J4。 开发板将启动，Z-Turn-Lite 上的蓝色 LED 将点亮，表示 PL 已成功配置，您将在 putty 终端上看到以下内容（您可能需要等待几秒钟才能使用完整的 Echo 服务器 输出显示在终端上）。

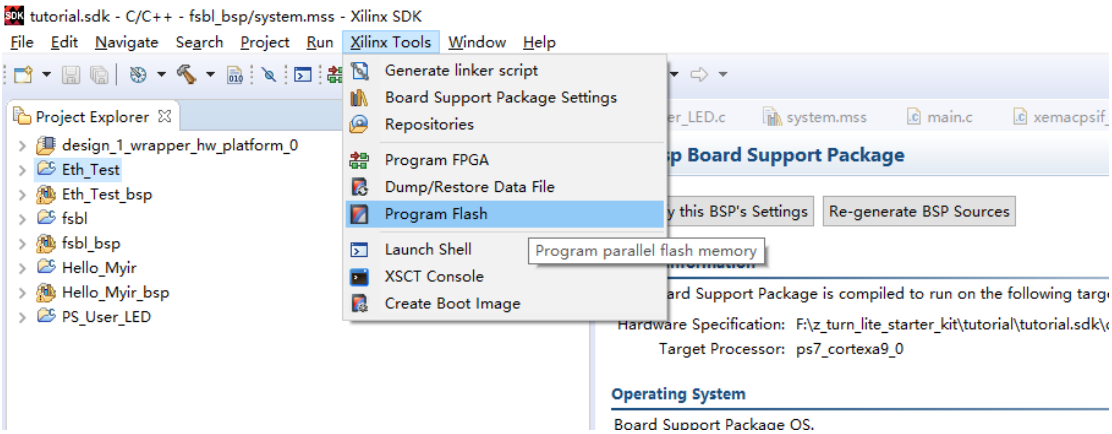


9.3 从 QSPI Flash 引导

为了从 QSPI Flash 引导，必须生成启动映像并将其烧写到 QSPI Flash 中。

9.3.1 使用引导映像对 QSPI Flash 进行编程

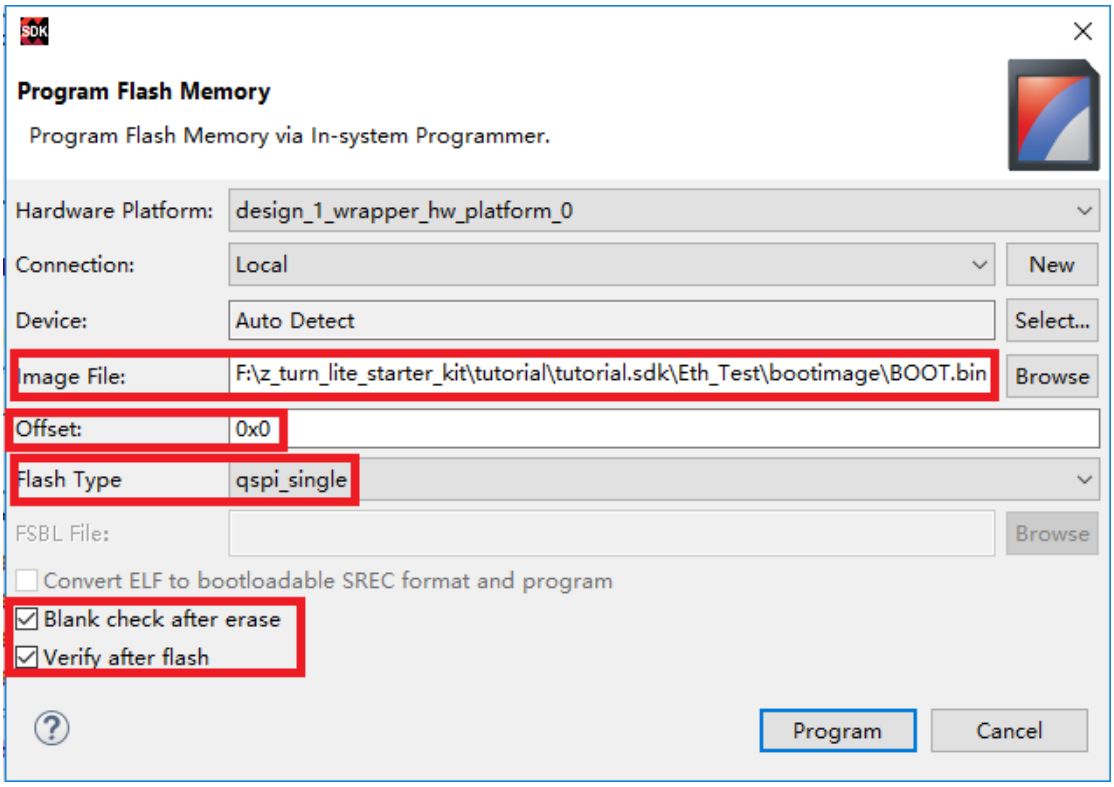
- 在开发板上的 J26 的 BT_JP2 和 BT_JP1 都安装跳线帽，
- 将 5v 电源连接到 z_turn_Lite 开发板的 J4 上。
- 在 SDK 工程管理器中，选择 **Xilinx Tools > Program Flash**



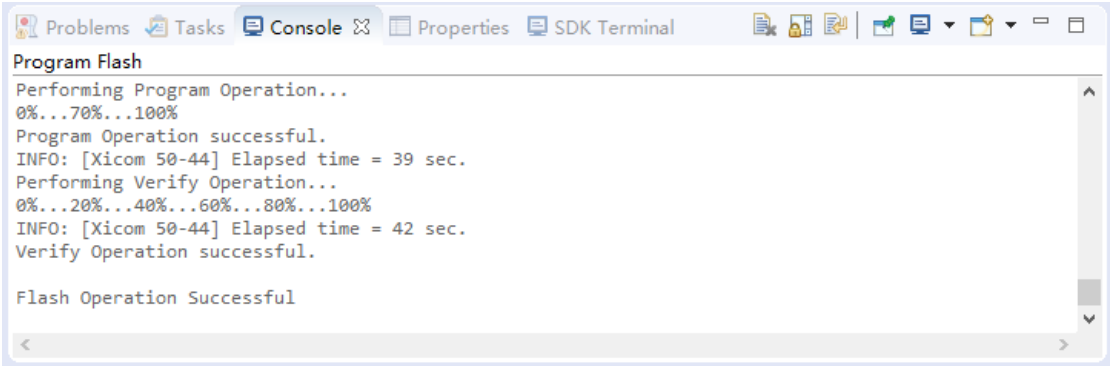
在出现的 Program Flash Memory 对话框

a. 按下图中的设置进行配置

b. 单击 Program 将程序烧写到 QSPI

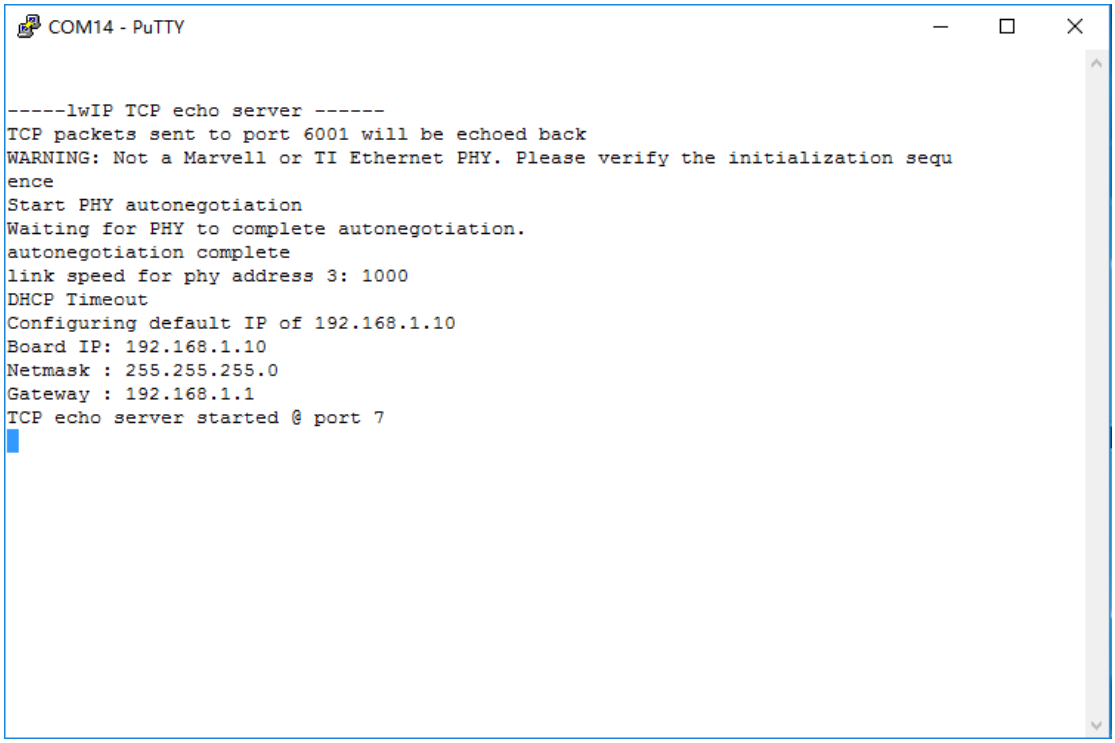


当烧写 QSPI 完成后出现如下图所示，显示 QSPI 已经烧写成功。

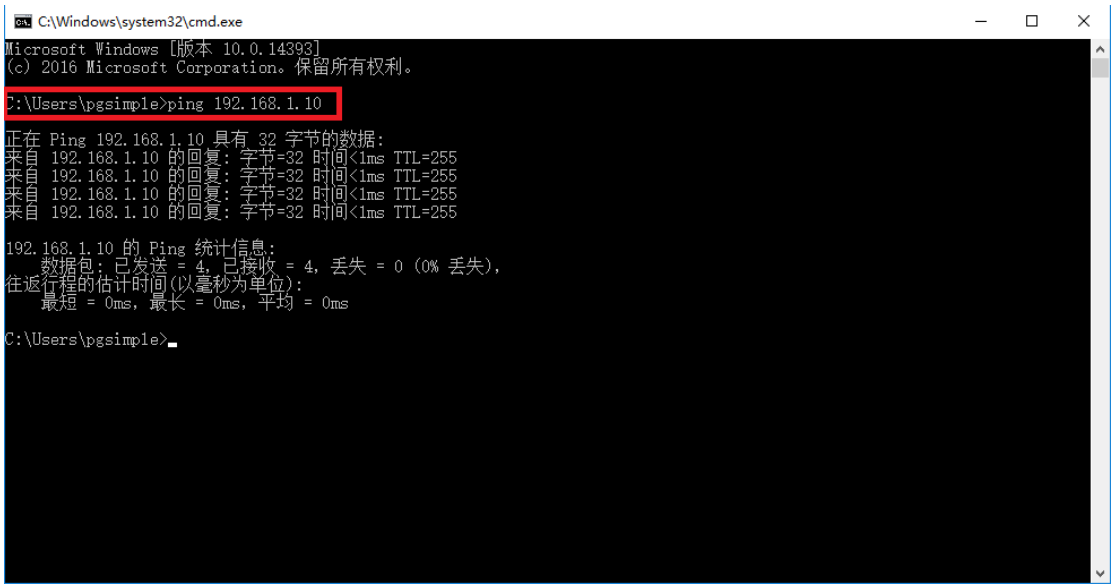


9.3.2 使用 QSPI Flash 启动开发板

•在开发板上的 J26 的 BT_JP2 和 BT_JP1 都安装跳线帽。
•将 5v 电源连接到 z_turn_Lite 开发板的 J4 上。



你现在可以像以前在下载时所做的那样 ping 这个开发板



10 电脑设置

本节描述如何在电脑上安装 USB 驱动程序，以便使用 USB-uart 连接到 z_turn_Lite 开发板上启动。

10.1 安装 USB UART 驱动程序

在计算机上安装 USB UART 驱动。

10.2 配置电脑的 COM 端口

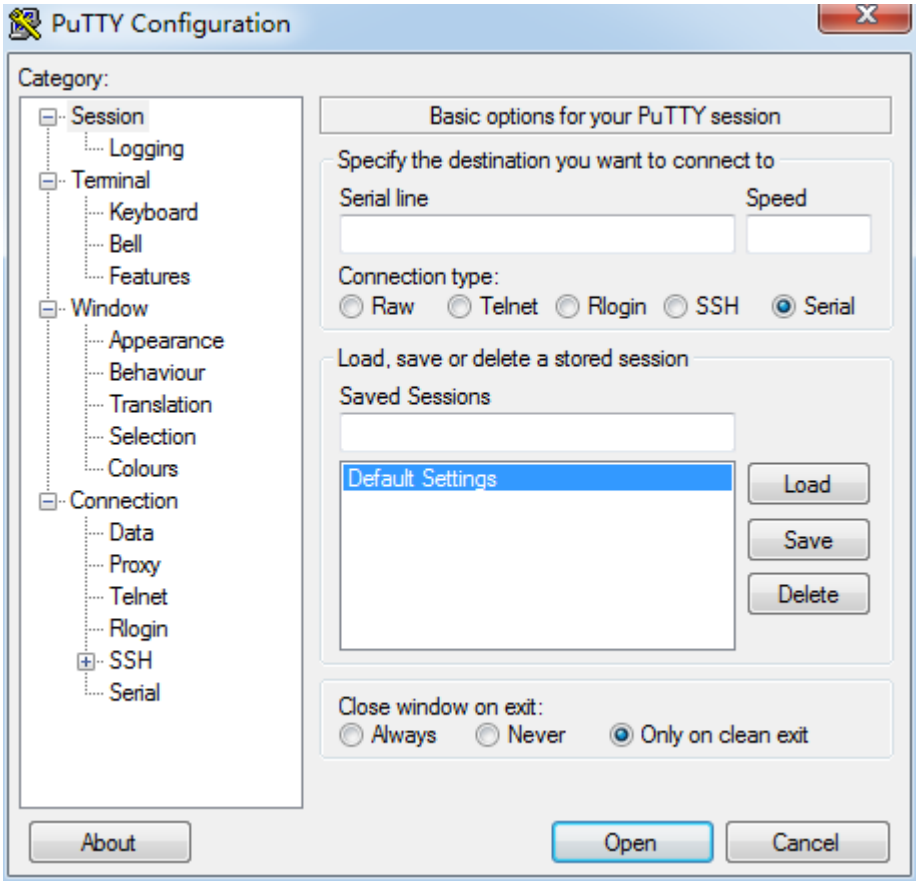
参考设计使用终端程序在主机和 Z-Turn-Lite 开发板之间进行通信。 为此配置主机 COM 端口：

- 通过 J11 USB-UART 端口将 Z-Turn-Lite 入门工具包连接到主机，并打开电源。
- 打开主机“设备管理”，如下图所示。 在 Windows 任务栏中，单击开始，单击控制面板，然后单击设备管理器。
- 扩展端口（COM&LPT），您将看到 USB 串行端口（COMXX）。 这是连接到 Z-Trun-Lite 的 COM。

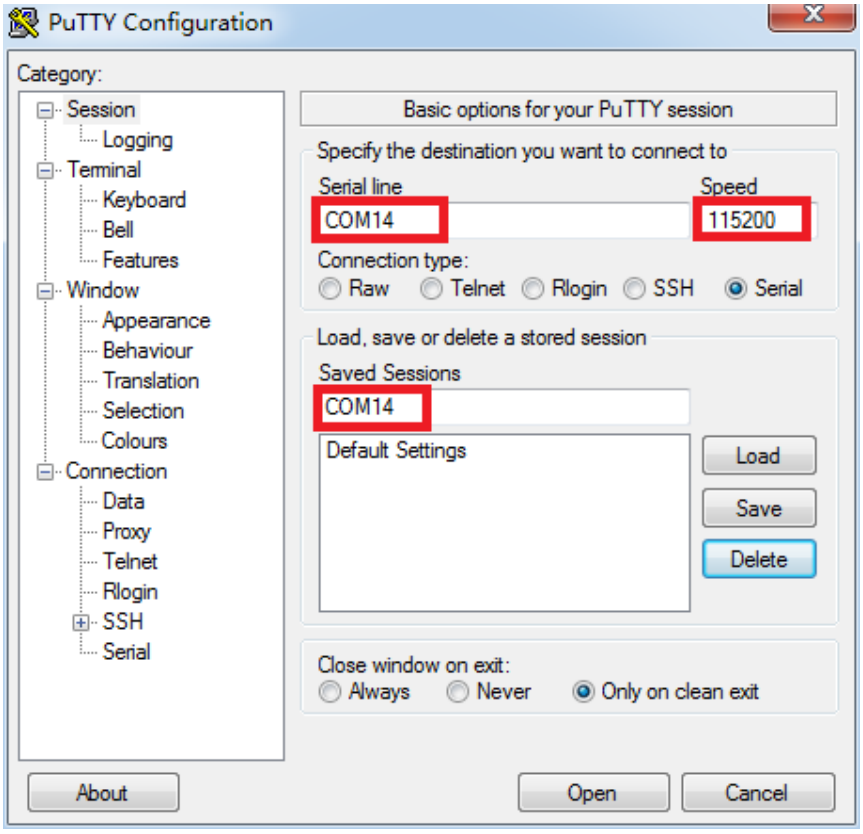
10.3 安装终端程序

将 putty 终端程序安装在主机上的磁盘 XX 中。 要与 Z-Turn-Lite 开发板进行通信，请配置新的连接和串行端口设置，如下图所示。 这些设置必须与上一节中显示的主机 COM 端口设置相匹配。

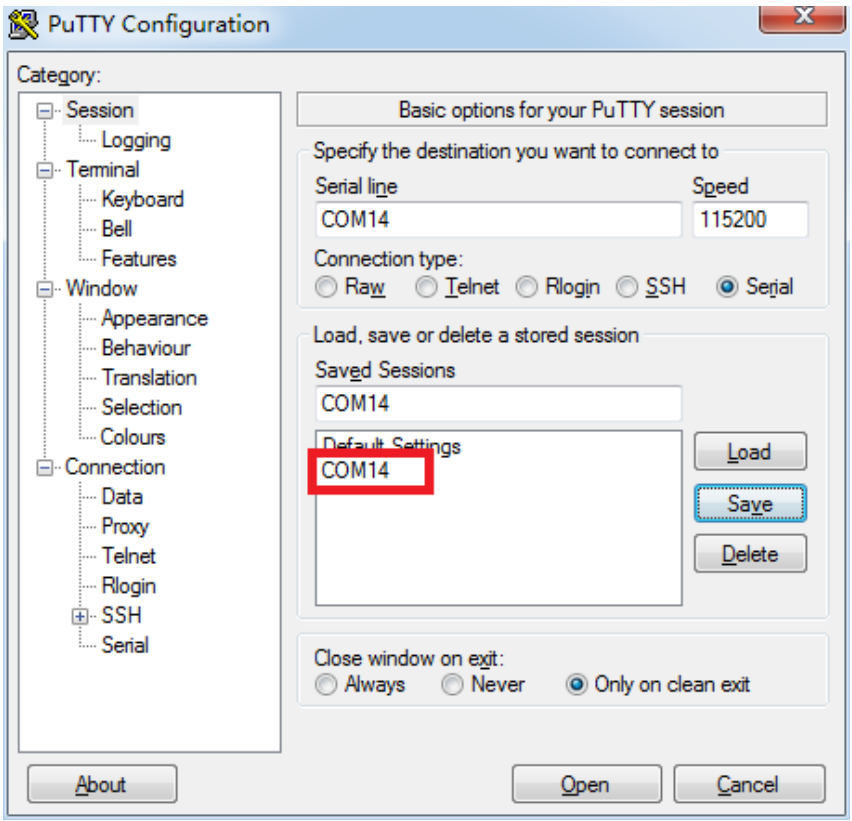
打开 putty 如下图所示



在 putty 中设置波特率为 115200，这个 COM 口要和设备管理器中的 COM 口一致这里为 COM14(不同的电脑 COM 口不一致根据自己的 COM 口自行设置)，设置方法是一样的。



点击 putty 中的 Save 就会看到 COM14 已经被添加进来了



选中 COM14，然后点击 Open 就可以打开 putty 调试界面了。