

Lab1: “Hello World”

——Xilinx EDK 设计基本流程

基于 Nexys 4 FPGA 平台

Lab 1: “Hello World”

实验简介

本实验旨在使读者学会 Xilinx 的 XPS 和 SDK 工具的使用，同时完成一个在串口上显示“Hello world!”的简单例程。

实验目标

在完成本实验后，您将学会：

- XPS 工具的使用流程，从新建工程到导入到 SDK。
- SDK 工具的使用流程，从导入到 SDK 到在板卡上运行 C 语言程序。

实验过程

本实验旨在指导读者使用 Xilinx 的 XPS 工具，调用串口的 IP 核，并将导入到 SDK，调用这个 IP 核，在串口上显示“Hello world!”，然后在 Nexys 4 平台上进行测试验证。

实验由以下步骤组成：

1. 在 XPS 中建立工程
2. 添加 IP 核并调整相关设置
3. 进行端口的互连
4. 将工程导入到 SDK
5. 在 SDK 中添加 c 语言源程序
6. 在 Nexys 4 上进行测试验证

实验环境

◆ 硬件环境

1. PC 机
2. Nexys 4 FPGA 平台

◆ 软件环境

Xilinx ISE Design Suite 14.3（FPGA 开发工具）

第一步 创建工程

1-1. 运行 Xilinx Platform Studio,创建一个空的新工程,基于 xc6slx45csg484-3 芯片和 VHDL 语言。

1-1-1. 选择 开始菜单 > 所有程序 > Xilinx Design Tools > ISE Design Suite 14.3 > EDK > Xilinx Platform Studio.点击运行 Xilinx Platform Studio(XPS) (Xilinx Platform Studio 是 ISE 嵌入式版本 Design Suite 的关键组件,可帮助硬件设计人员方便地构建、连接和配置嵌入式处理器系统,能充分满足从简单状态机到成熟的 32 位 RISC 微处理器系统的需求。)

1-1-2. 点击 **Create New Project Using Base System Builder** 来打开新工程建立向导。会出现一个 **Create New XPS Project Using BSB Wizard** 对话框,如图 3。



Getting Started



Documentation



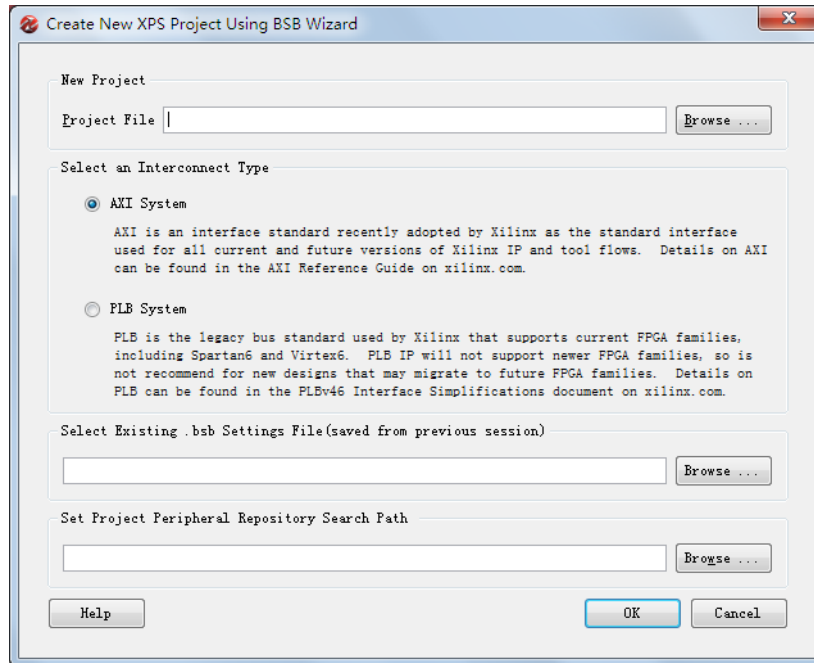


图 3：新工程建立向导

- 1-1-3. 如图 3，在新工程建立向导对话框的 **Project File** 栏选择工程建立后存放的路径，这里可以选择 **c:\Nexys4_lab**，可以将 **system.xmp** 改成所建立工程的名字，这里取 **lab1**(名字中不要有中文和空格)，于是 **Project File** 栏中的路径变为 **c:\Nexys4_lab\lab1\lab1**。点击 **OK**。
- 1-1-4. 新出现的是关于工程的一些参数设置的对话框，设置如下的参数后，点击 **Next**，如图 4。
- architecture: artix 7**
 - Device: XC7a1007**
 - Package: CSG324**
 - Speed: -3**

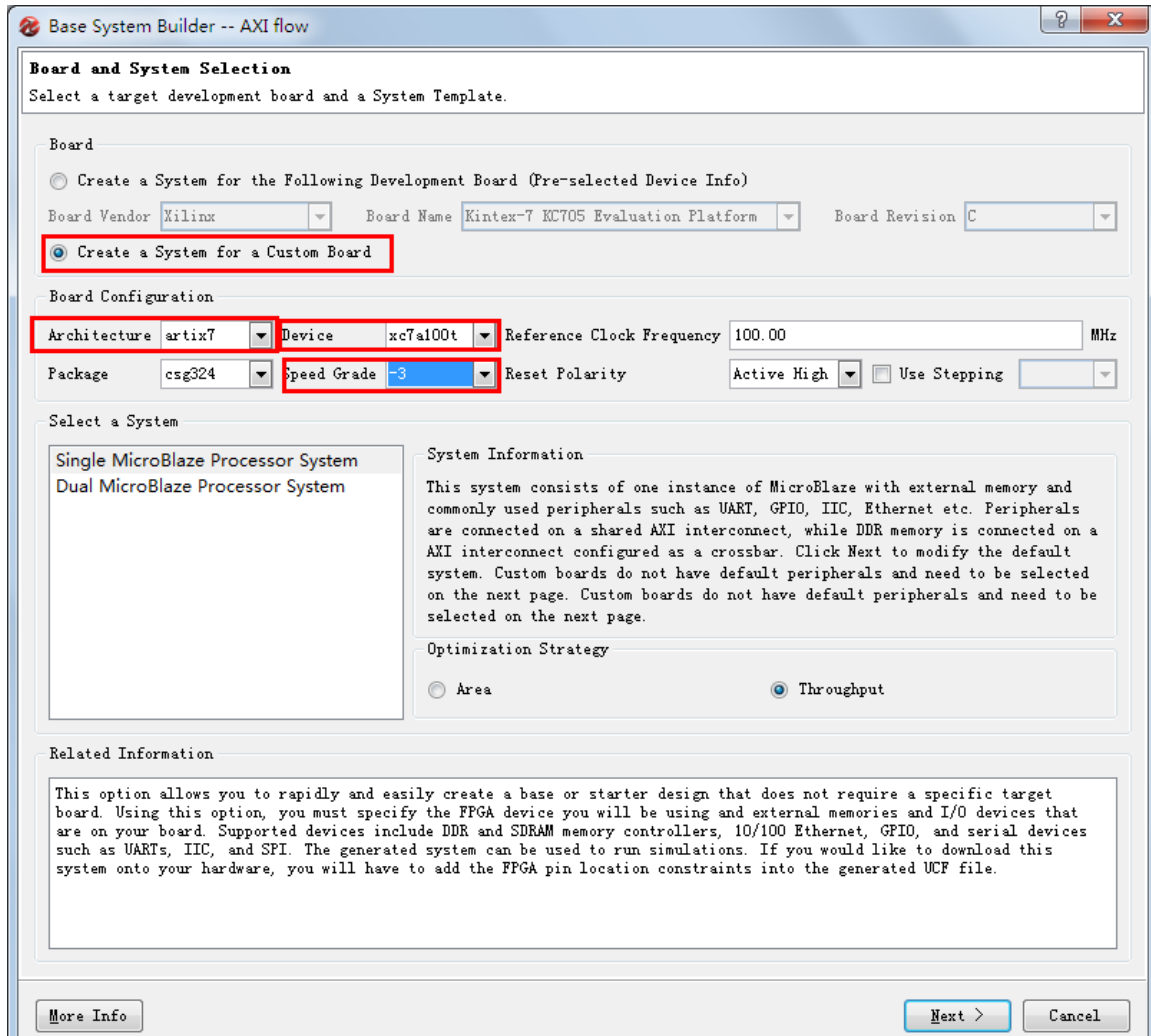


图 4：新工程参数设置

1-1-5. 在接下来出现的页面中选择要添加的 IP 核，并设置 IP 核的参数：

单击 Select and configure Peripherals 下的 Add Device...

出现图 5 中的蓝色对话框。

在 IO Interface Type 中的下拉菜单中选择 UART。

在 Device 的下拉菜单中选择 RS232。

单击 OK

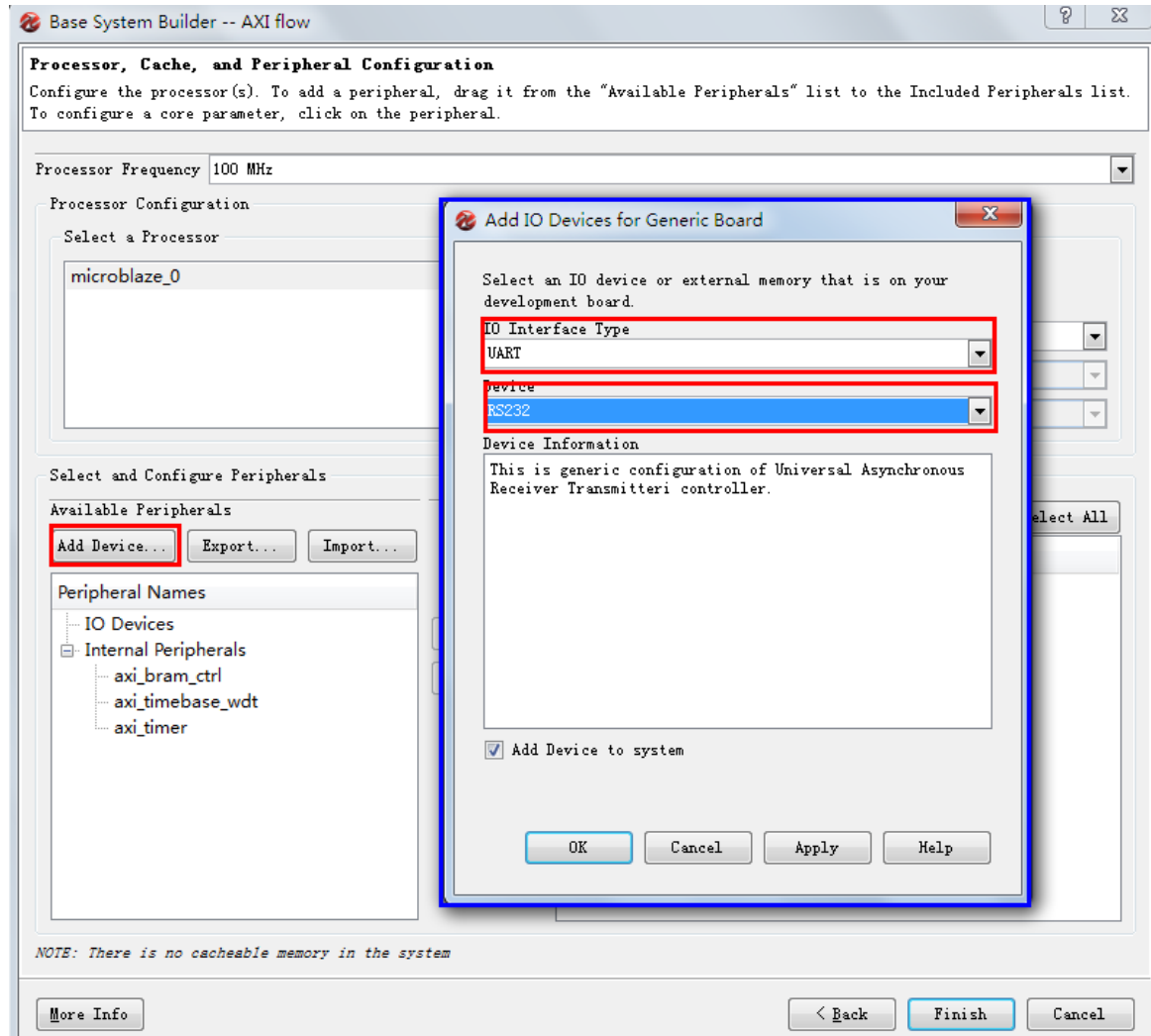


图 5. 添加串口的 IP

- 1-1-6. 注意,串口的默认波特率设置为 9600。在 Lab1 中, 不做修改。以后的设计中根据需要进行调整。但是为了确保串口的正常通讯, SDK 工程中的 Terminal 的波特率以及串口的其他设置必须与之保持一致。

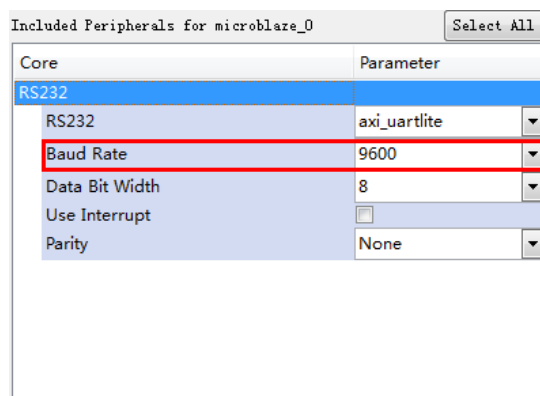


图 6. 串口的设置

第二步 进行端口的互连

2-1. 在 PORT 选项卡中修改时钟的相关设置

2-1-1. Port 选项卡（展开 External Port），如图 7.

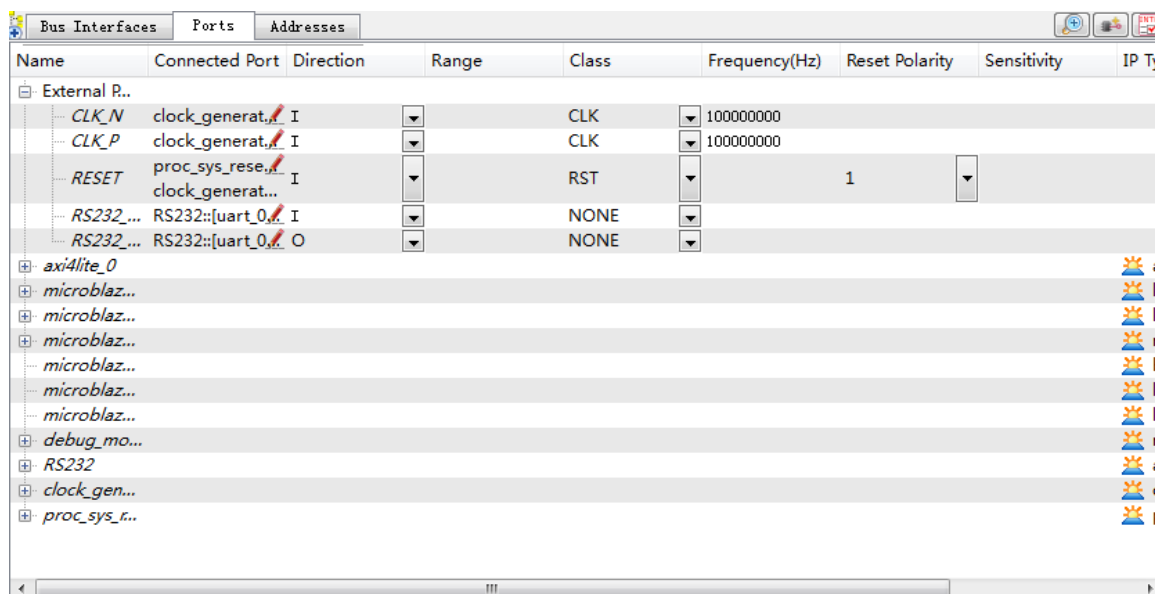


图 7: PORT 选项卡的初始状况

2-1-2. 将 External Port 中的 CLK_N 和 CLK_P 都去掉。

右键选中该端口，然后点击 Delete External Port，如图 8.

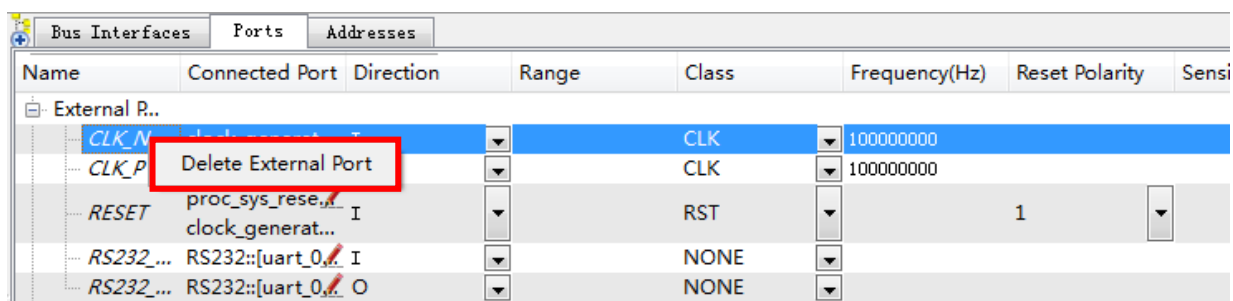


图 8: 源文件添加后的 ISE 用户界面

2-1-3. 将 **Clock_generator_0** 作为新的时钟，加入外部端口。

找到 **Clock_generator_0** 中的 **CLKIN**，右键选中，在菜单中点击 **Make external**

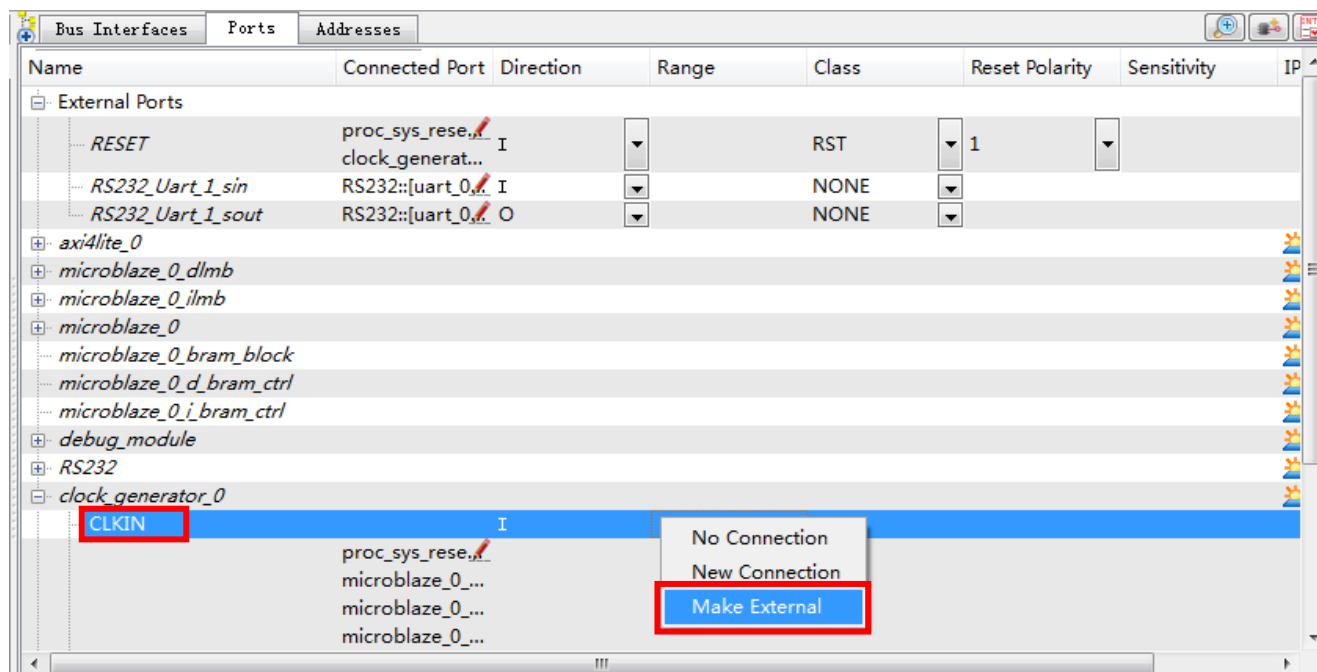


图 9: Clock_generator_0 中的 CLKIN

注意 **External** 中的 **Name** 一项，这是我们添加用户约束文件（UCF）的依据。

Name	Connected Port	Direction	Range	Class	Reset Polarity	Se
External Ports						
RESET	proc_sys_res...	I		RST	1	
RS232_Uart_1_sin	RS232::[uart_0...	I		NONE		
RS232_Uart_1_sout	RS232::[uart_0...	O		NONE		
clock_generator_0_CLKIN_pin	clock_generat...	I		CLK		

图 10: 修改后的 External PORT

第三步 添加用户约束文件

3-1. 打开初始 **UCF** 文件，根据需求进行修改

3-1-1. 在页面偏左找到 **IP catalogue / Project** 选项卡，双击 **UCF File: DATA\lab1.ucf**，ucf 文件在右侧打开

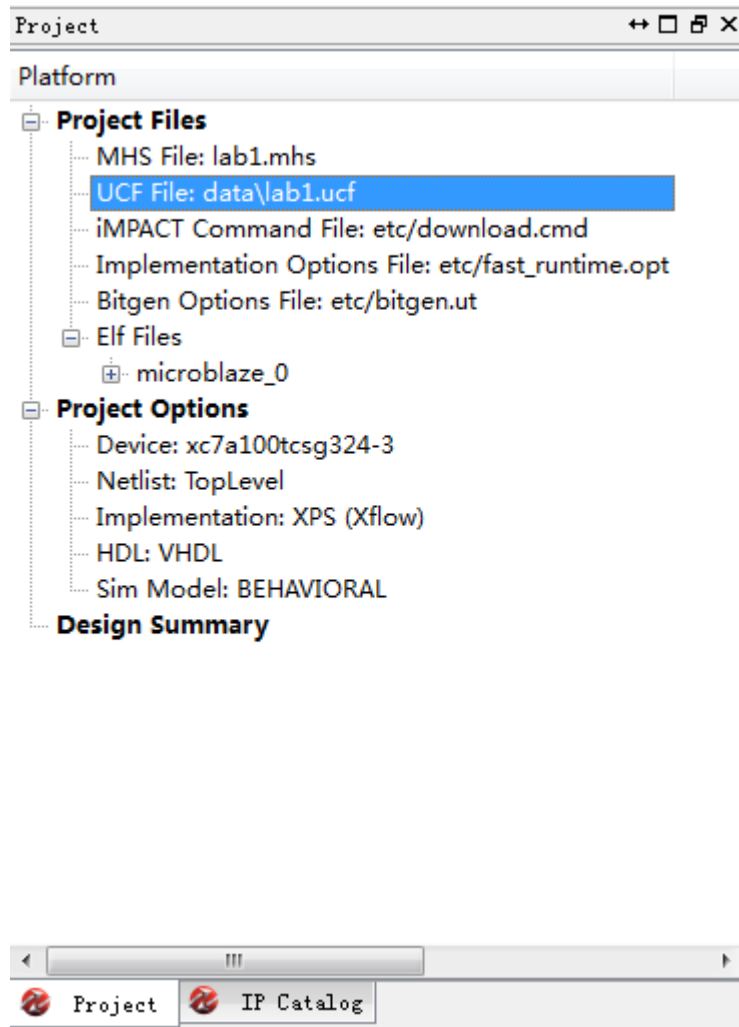


图 11: UCF 文件的位置

3-1-2. 这里我们手动输入 LOC（引脚位置）约束代码，如图 12。点击保存。

在编写完下图用户约束文件引脚约束代码之后，XPS 设计中的那些外部端口就连接到了 FPGA 芯片的相关引脚上，从而与 Nexys 4 板上的外设联系起来。在这里，我们将 RESET 信号与一个 switch 相连。其实也可以选用一个 Button。在不确定复位信号是高有效还是低有效的时候，用 switch 优于 button。因为如果复位信号低有效，并且用 button 当复位，为了让系统运行，就要一直摁住 button。RS232 的发送和接收，则与板子上的 C4 和 D4 管脚相连。

```
## Clock signal
NET "clock_generator_0_CLKIN_pin" LOC = "E3" | IOSTANDARD = "LVCMOS33";
NET "clock_generator_0_CLKIN_pin" TNM_NET = sys_clk_pin;
TIMESPEC TS_sys_clk_pin = PERIOD sys_clk_pin 100 MHz HIGH 50%;

## Switches
NET "RESET" LOC = "U9" | IOSTANDARD = "LVCMOS33";

## USB-RS232 Interface
NET "RS232_Uart_1_sin" LOC = "C4" | IOSTANDARD = "LVCMOS33";
NET "RS232_Uart_1_sout" LOC = "D4" | IOSTANDARD = "LVCMOS33";
```

图 12: UCF 文件

3-1-3. 保存之后将工程导入到 SDK

在页面左边，点击 **Export Design**，如图 13。

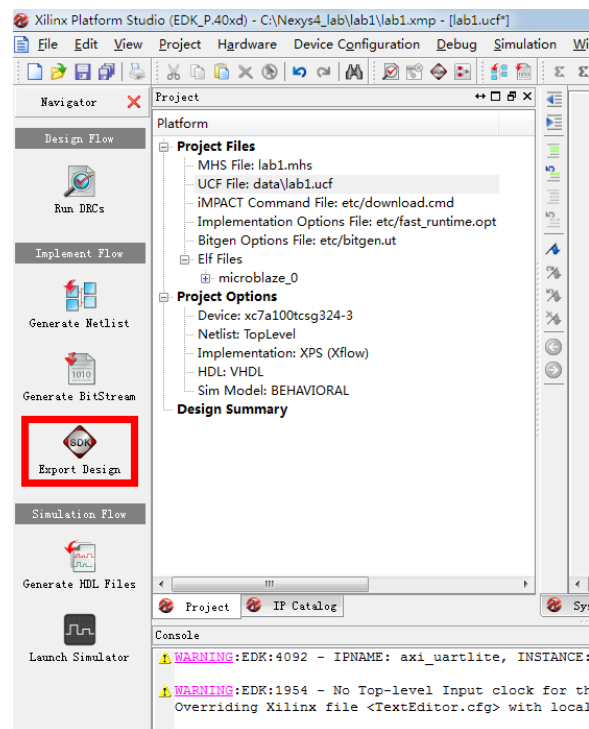


图 13: export design

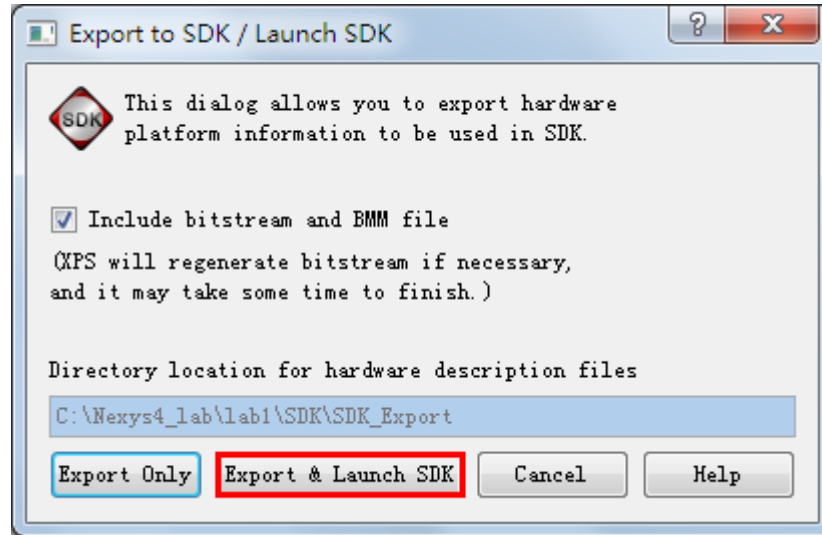


图 14：在弹出的对话框中选择 **Export & launch sdk**

3-1-4. 选择 SDK 导入路径

注意要具体到..\sdk\sdk_export，如图 14

点击 **ok**

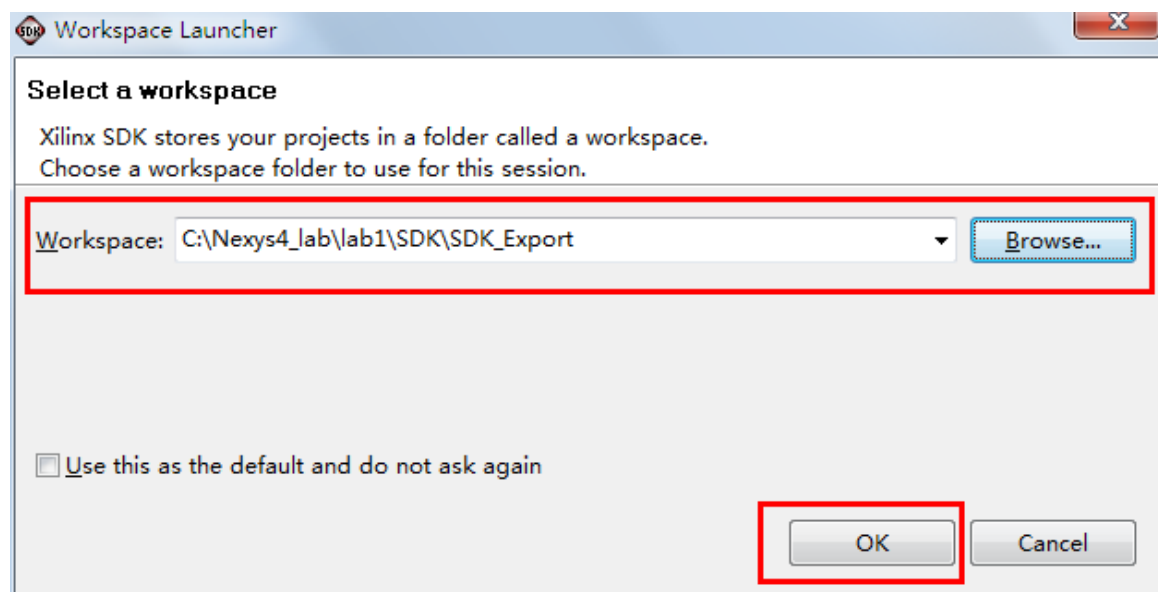


图 14：导入 **sdk** 的工作空间选择

4-1. 添加软件应用。

4-1-1. 在 SDK 的用户界面中，选择 **file—new—application project**，如图 15。

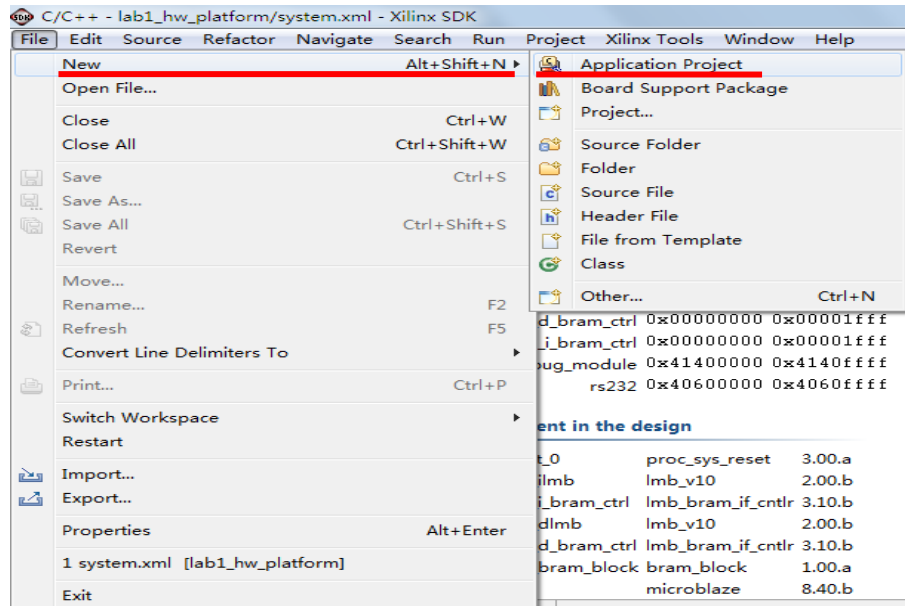


图 15: 新建软件应用

4-1-2. 输入工程的名称，这里使用 **helloworld_test**，同样不要包含空格和中文，点击 **next**

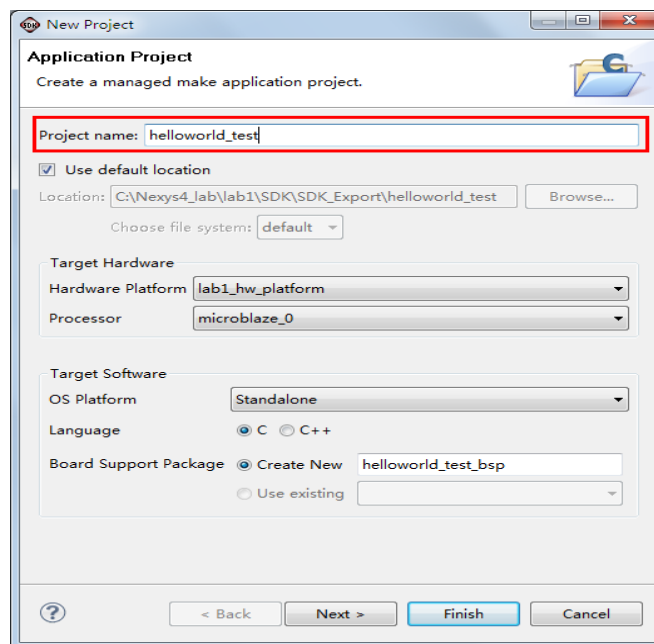


图 16: 新建工程---命名

4-1-3. 在下一步弹出的对话框中选择 **hello world**，然后点击 **finish**

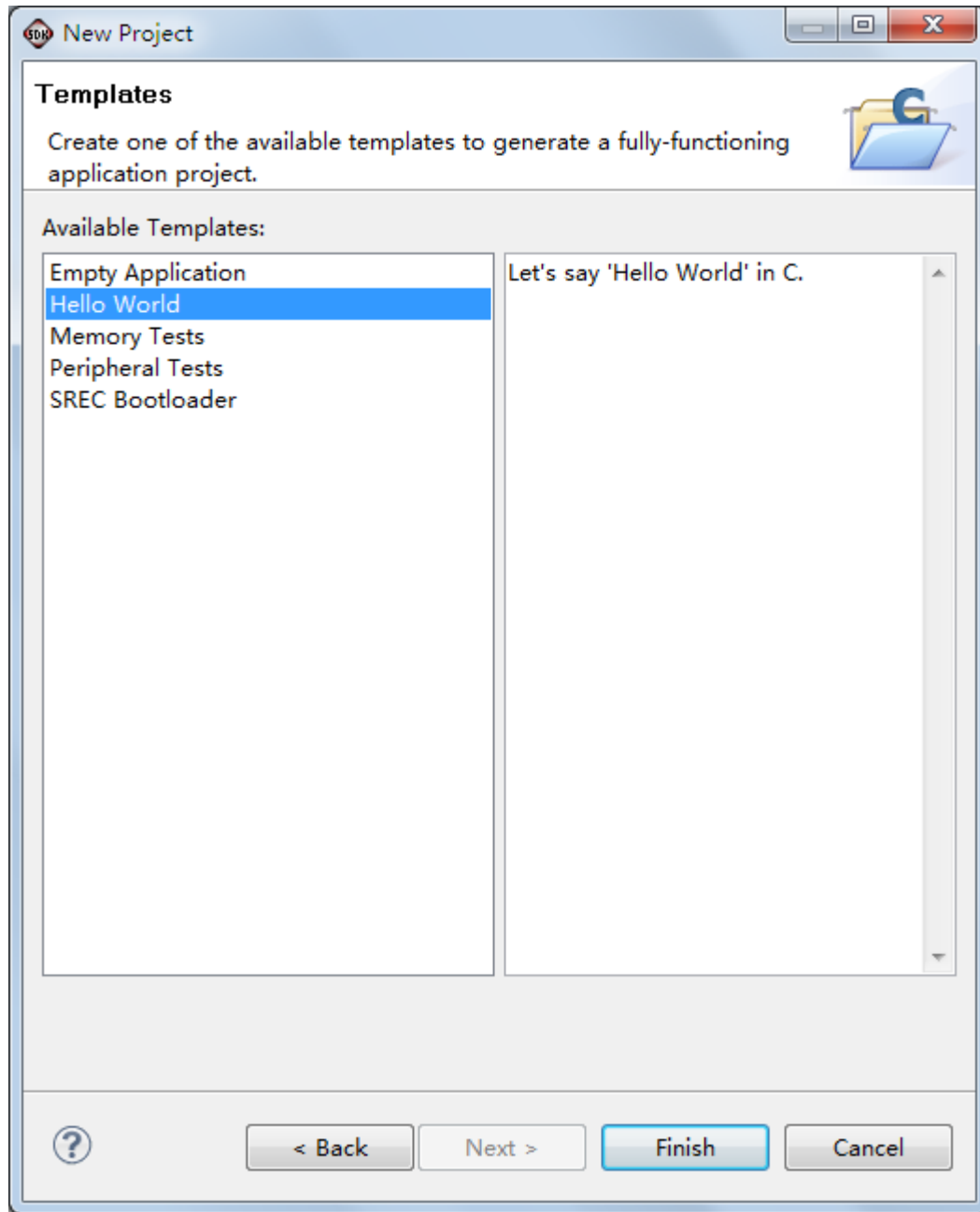


图 16: 选择 Helloworld 示例代码

4-1-4. 添加完毕以后可以在左侧双击源文件，查看这段代码：

可以根据自己的需要进行一些修改，修改后保存。

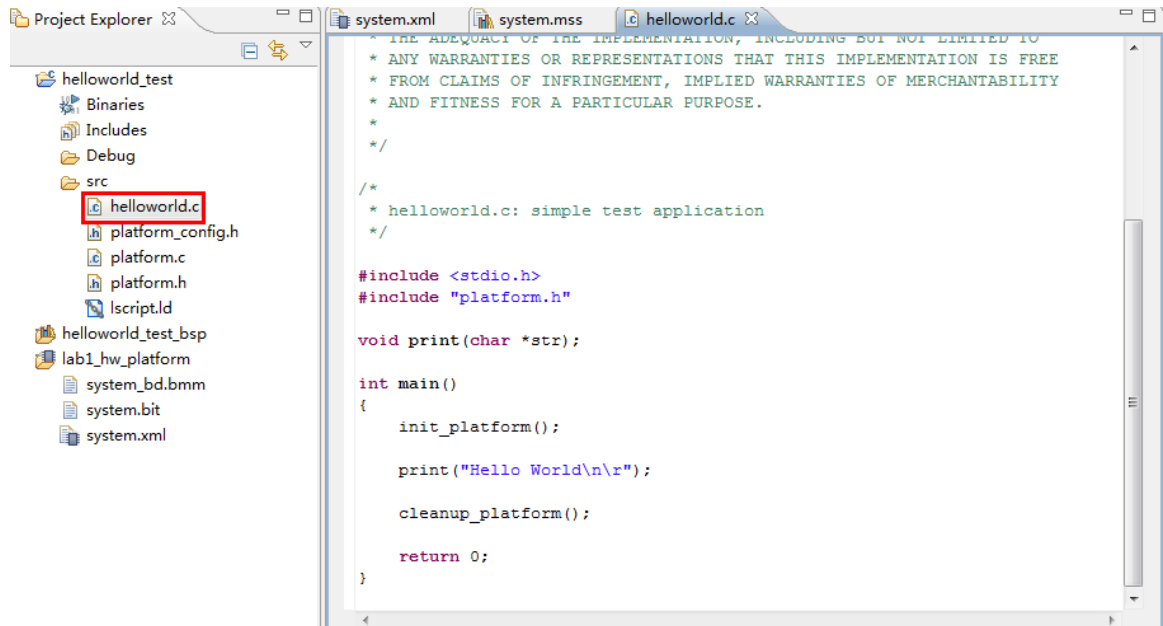


图 17: helloworld 源代码

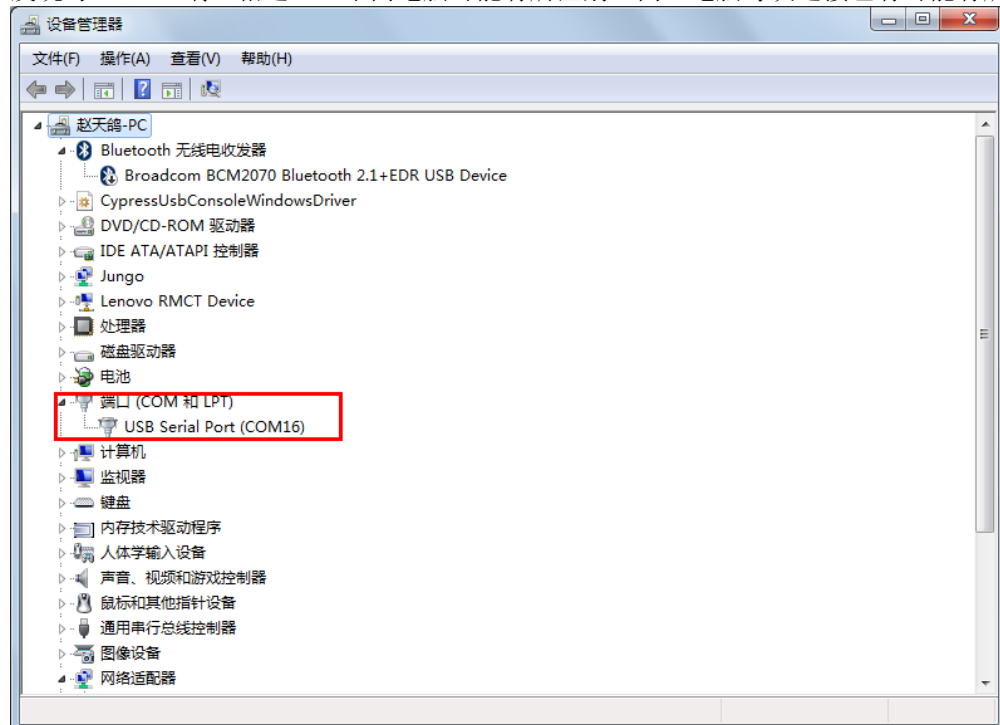
第五步 上板验证

5-1. 将 Nexys4 与 Pc 的 USB 接口连接

5-2. 查看端口号:

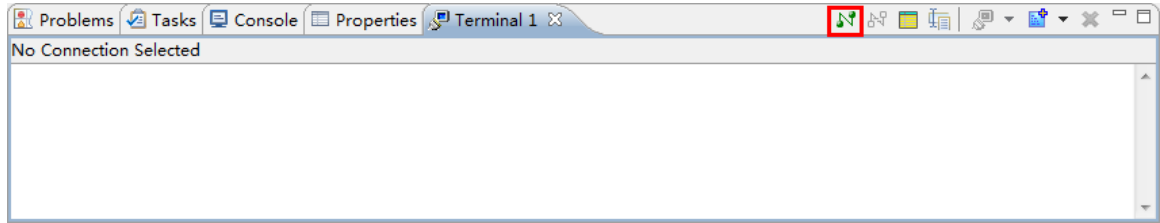
右键“我的电脑”——“属性”——在页面左侧选择“设备管理器”

发现与 com16 端口相连: (不同电脑可能有所区别, 同一电脑每次连接也有可能有所区别)

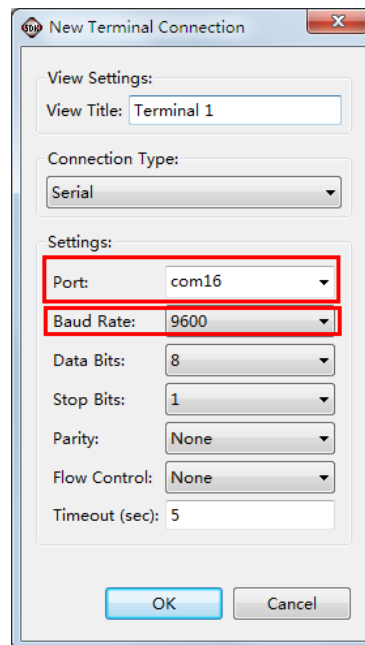


5-3. 在 SDK 中打开串口：

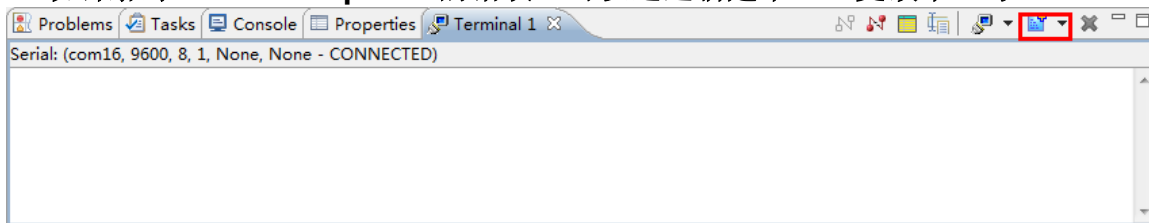
5-3-1. 在下面的在页面下方找到 **terminal** 选项卡，然后点击绿色的连接按钮。



5-3-2. 按照端口号和 XPS 中的波特率（**baud rate**）进行如下设置：

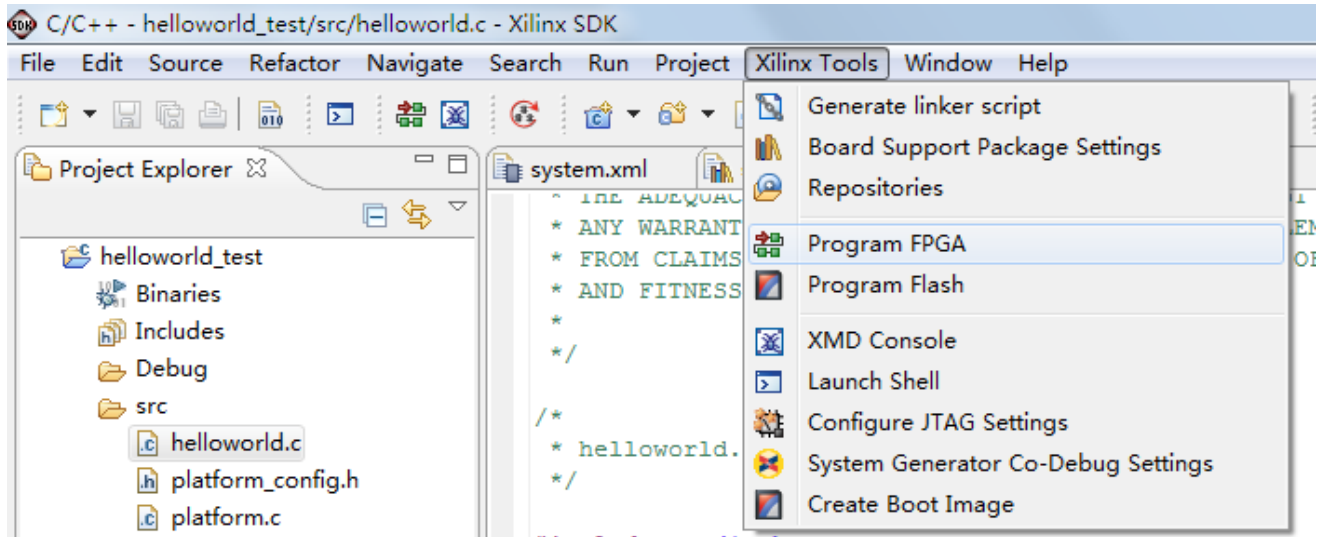


如果报了“no such port”的错误，可以通过新建串口，更改串口号：

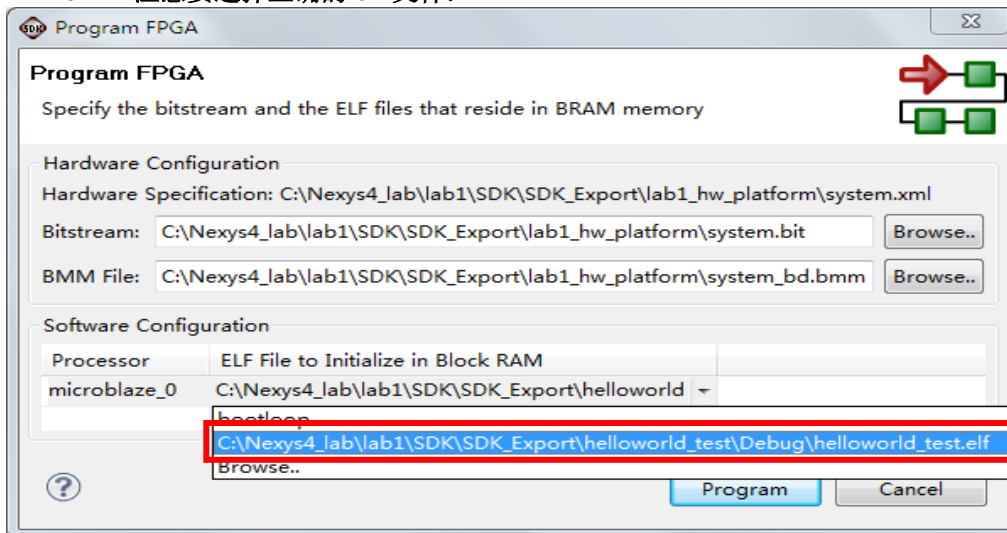


5-4. 将程序下载到板子上并运行

5-4-1. 在页面上方，**xilinx tools** 下拉菜单中选择 **program fpga**



5-4-2. 注意要选择正确的 elf 文件:



点击 program

5-5. 在串口中看到结果:

