25A, No.1369,Dongfang Road, Shanghai
Tel:+86-021-58893151 Fax: +86-021-58894598

Lab11: USB 接口实验

*基于 **Nexys 4 FPGA** 平台*

Lab11 : USB 接口实验

实验简介

本实验旨在使读者进一步熟悉 Xilinx 的 XPS 和 SDK 工具的使用，并掌握 PS2、UART、PLB、AXItoPLB BRIDGE 这四个 ip 核的添加方法。

实验目标

在完成本实验后，您将学会：

- 如何在 XPS 工具中添加并根据设计需求调整 PS2、UART、PLB、AXItoPLB BRIDGE 这四个 IP 核。
- 如何通过外接 USB 键盘实现 PS2 的通信并在串口观察现象。

实验过程

本实验旨在指导读者使用 Xilinx 的 XPS 工具，调用 PS2、UART、PLB、AXItoPLB BRIDGE 这四个 IP 核，并将导入到 SDK，调用它们，通过在串口进行对程序的调试，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。），**如图 1 所示。**

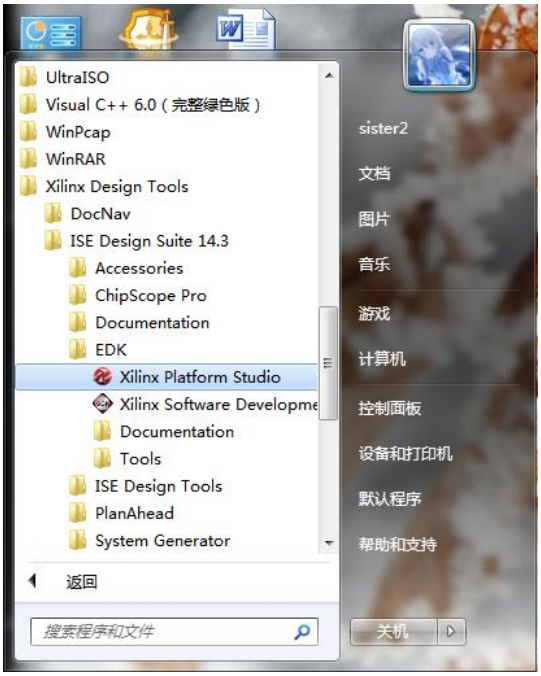


图 1：XPS 软件打开位置

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 2**。

图 2：新工程建立界面

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，**大家可以建立任意路径，只要路径名称不包含中文及空格即可。**建立的工程的名称我们使用默认的 system.xmp 即可。点击 **OK**。

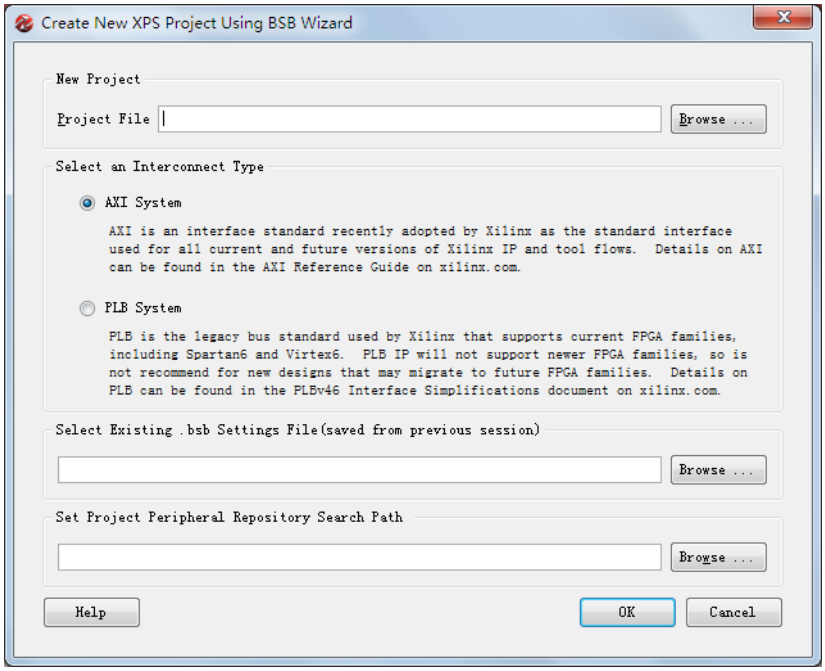


图 3：新工程建立向导

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: artrix 7
Device: XC7a1007
Package: CSG324
Speed: -3

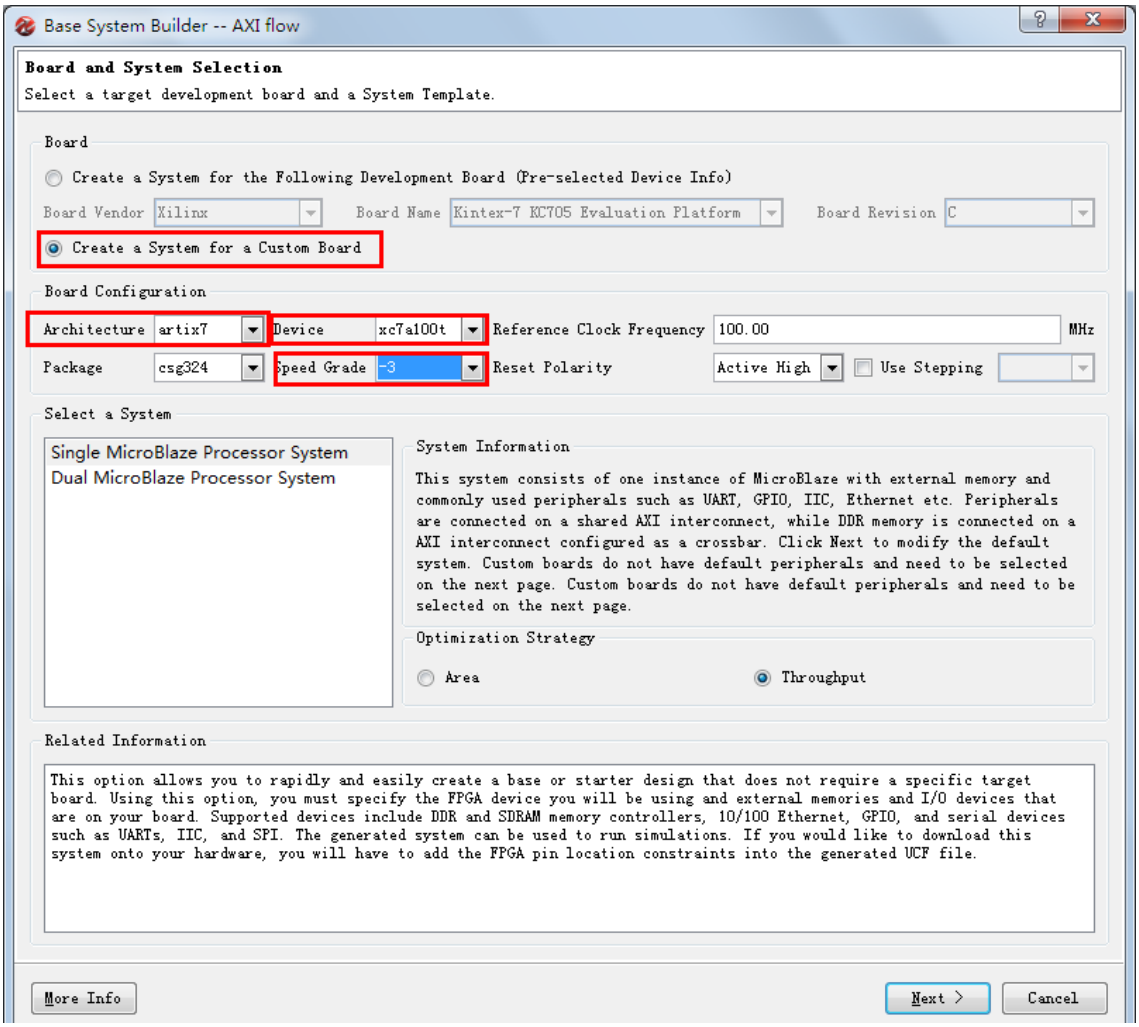


图 4：新工程参数设置

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**，即可添加 **UART** 的 **IP** 核。

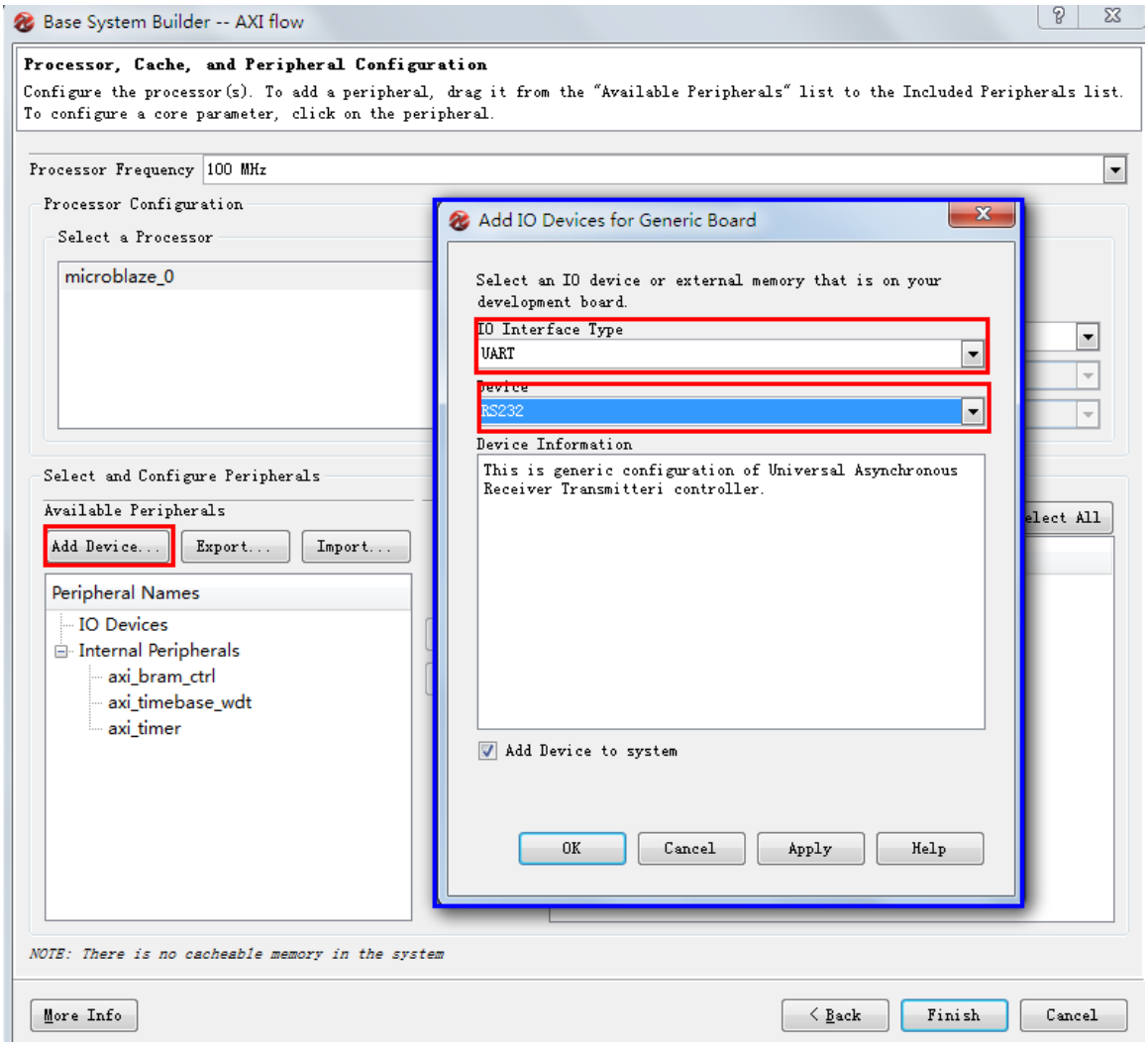


图 5：添加串口的 IP

1-1-6. 注意,串口的默认波特率设置为 9600。在 Lab2 中，我们需要统一修改到 115200，以便提高数据传输速度，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。

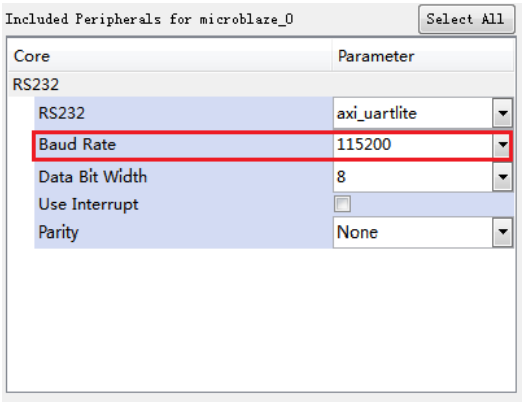


图 6：串口的设置

1-1-8. 下面我们要更改 FPGA 内部存储器 BRAM 的大小，将它的大小改为 64K，以防止存储空间太小无法存储软件代码部分。

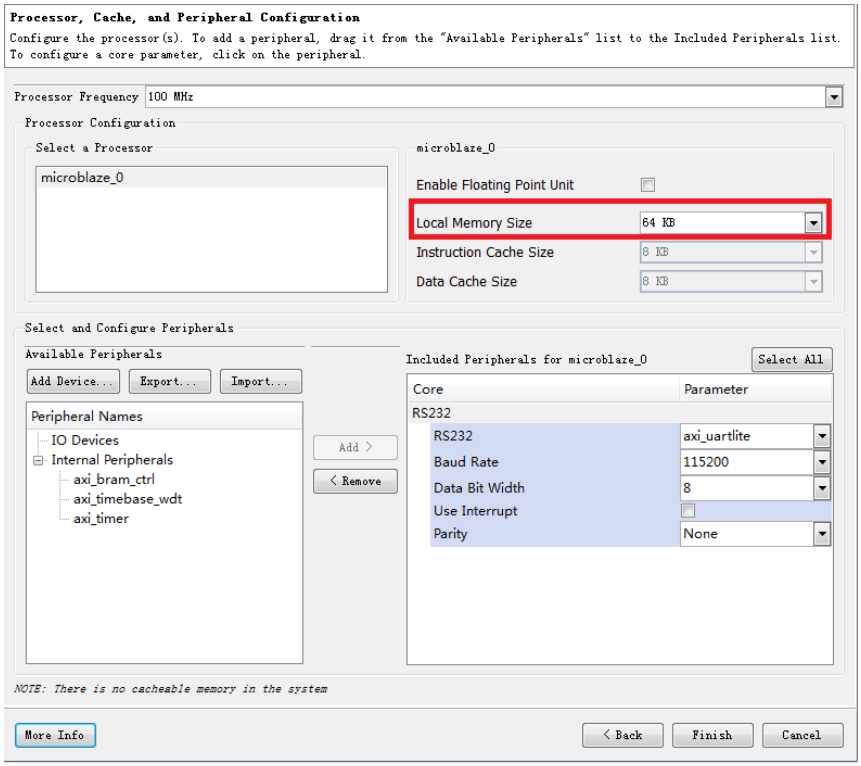
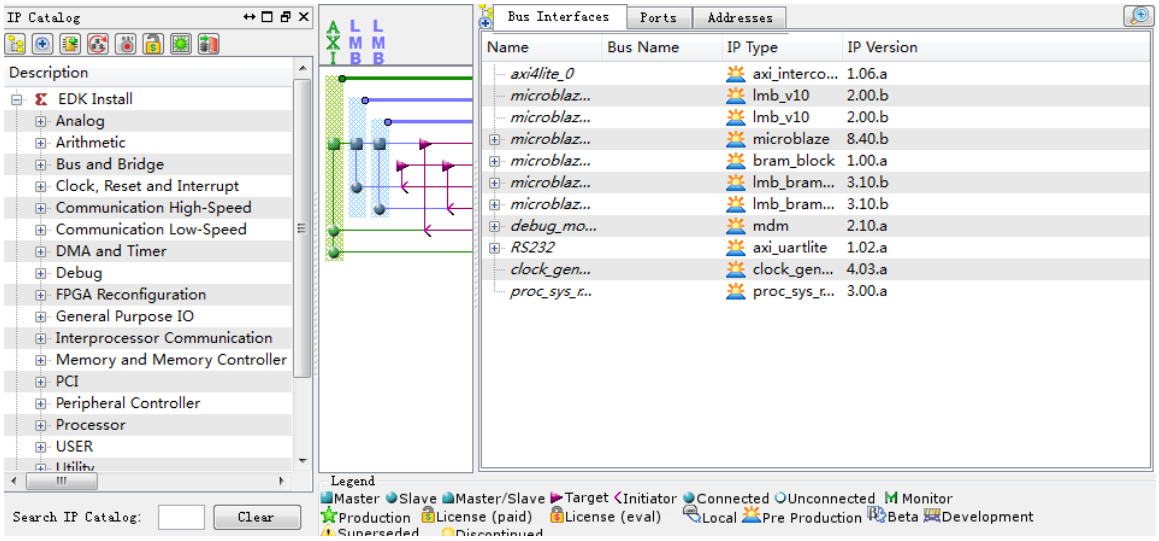


图 7：修改 local memory 大小

点击 finish。至此，包含串口 IP 核（RS232）的 EDK 工程就创建完成。如下图所示



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 8.

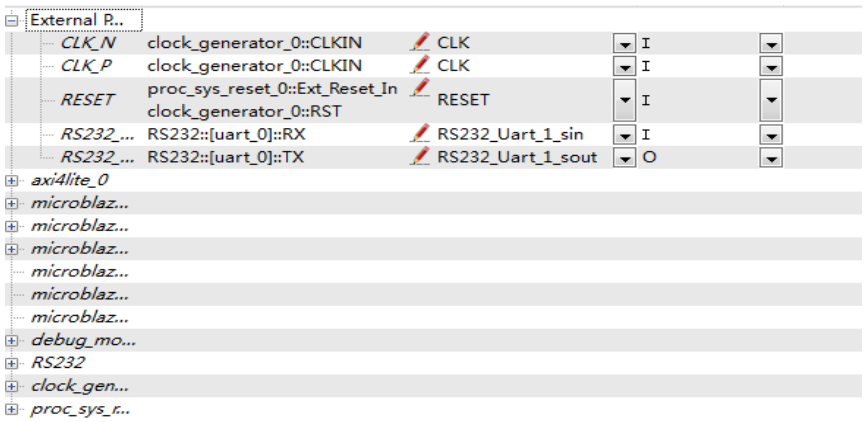


图 8：PORT 选项卡的初始状况

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 9。

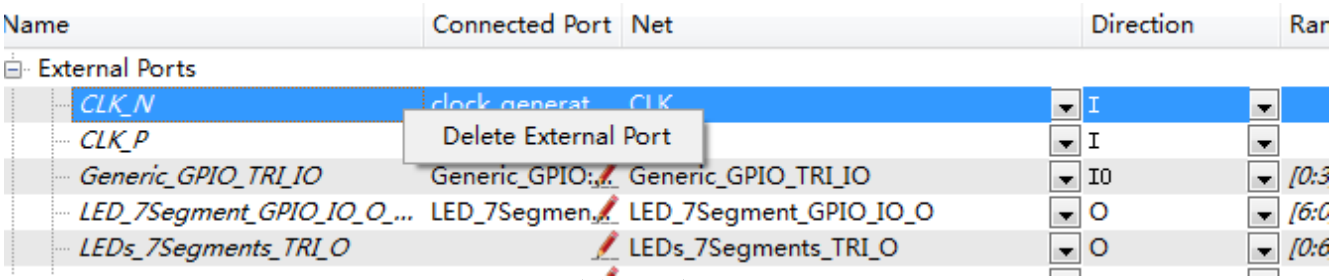


图 9：删除外部端口

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**

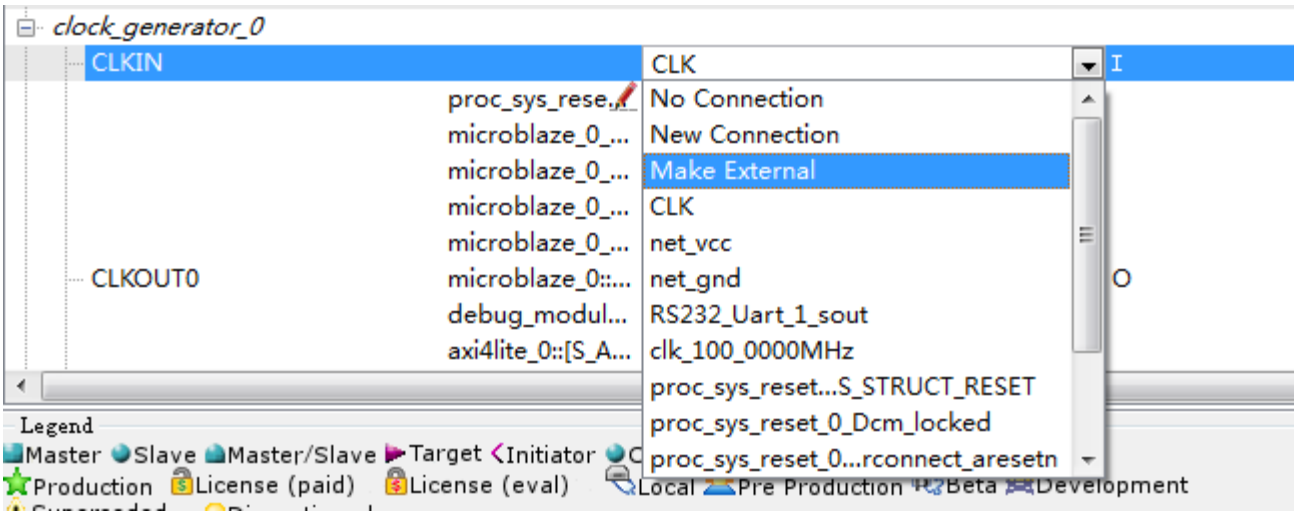


图 10： Clock_generator_0 中的 CLKIN

2-1-4. 下面我们要加入 3 个要用到的 IP 核，因为 xilinx 没有提供在 AXI 总线下的 PS2 ip 核，只提供了 PLB 总线下的，所以首先我们先加入 PLB 总线的 IP 核，之后再添加 PS2IP 核。我们先点击菜单栏的 IP CATALOG 按钮，点击 BUS AND BRIDGE，双击添加 PLB 总线。

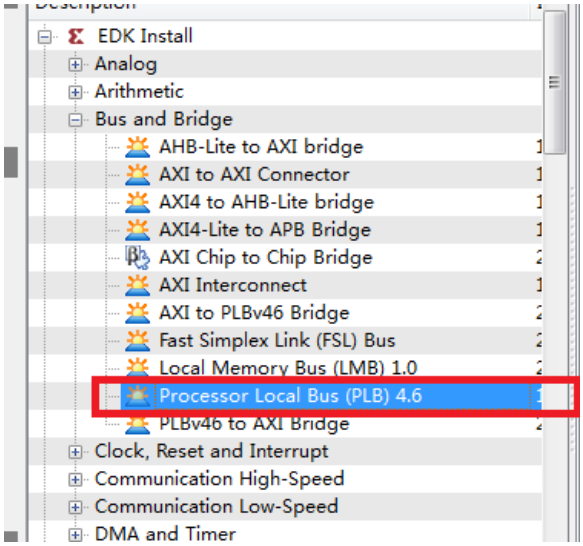
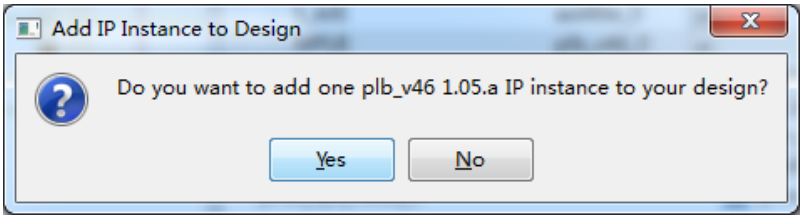
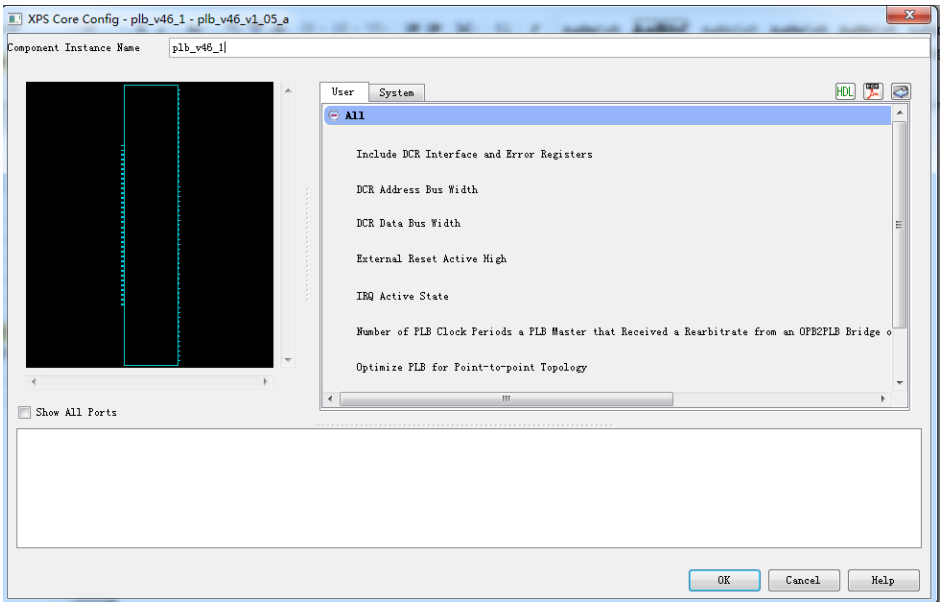


图 11：添加 PLB 总线

双击完毕之后点击 YES，之后点击 OK。





2-1-5. **我们要添加 AXI 总线到 PLB 总线的 BRIDGE 来连接两根总线，点击 PROJECT 界面的 BUS AND BRIDGE，双击 AXITOPLBv46 BRIDGE 来添加该 IP。**

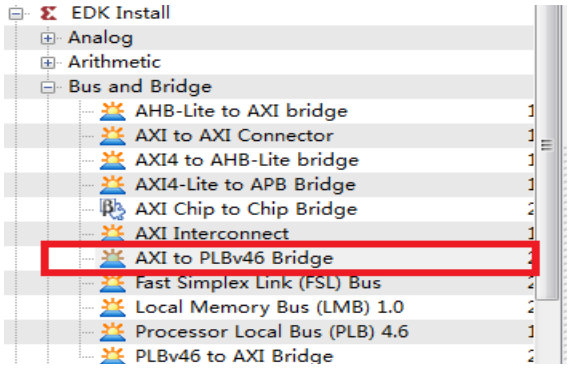


图 12：添加 BRIDGE

按照 2-1-4 中的步骤添加 IP 核。

2-1-6. 我们要添加 PS2 的 IP 核，由于 PS2 的 IP 核并不会自动识别 artix7 芯片的板卡，所以我们要手动添加 PS2 的 IP 核信息。

注意：我们先来到 xilinx 软件安装路径下的 \ISE14.3\14.3\ISE_DS\EDK\hw\XilinxProcessorIPLib\pcores 路径，找到里面的 xps_ps2_v1_01_b 这个文件夹，将这个文件夹拷贝到你工程目录下 pcores 文件夹，打开里面 DATA 文件夹下的后缀为 MPD 的文件，在里面添加 artix7=PRODUCTION 这句话并保存。

```
OPTION HDL = VHDL
OPTION DESC = XPS PS2 Interface
OPTION LONG_DESC = PLBV46 to PS2 Adapter
OPTION IP_GROUP = Communication Low-Speed:MICROBLAZE:PPC
OPTION ARCH_SUPPORT_MAP = (aspartan3=PRODUCTION, spartan3=PRODUCTION,
spartan3an=PRODUCTION, spartan3a=PRODUCTION, spartan3e=PRODUCTION,
spartan3adsp=PRODUCTION, virtex4lx=PRODUCTION, virtex4sx=PRODUCTION,
virtex4fx=PRODUCTION, virtex5lx=PRODUCTION, virtex5sx=PRODUCTION,
virtex5fx=PRODUCTION, aspartan3e=PRODUCTION, aspartan3a=PRODUCTION,
aspartan3adsp=PRODUCTION, qvirtex4lx=PRODUCTION, qvirtex4sx=PRODUCTION,
qvirtex4fx=PRODUCTION, qrvirtex4lx=PRODUCTION, qrvirtex4sx=PRODUCTION,
qrvirtex4fx=PRODUCTION, spartan6t=PRODUCTION, spartan6=PRODUCTION,
spartan6l=PRODUCTION, qspartan6t=PRODUCTION, qspartan6=PRODUCTION,
aspartan6t=PRODUCTION, aspartan6=PRODUCTION, virtex6lx=PRODUCTION,
virtex6sx=PRODUCTION, virtex6cx=PRODUCTION, virtex6llx=PRODUCTION,
virtex6lsx=PRODUCTION, qspartan6l=PRODUCTION, qrvirtex5=PRE_PRODUCTION,
qvirtex6lx=PRODUCTION, qvirtex6sx=PRODUCTION, qvirtex6fx=PRODUCTION,
qvirtex6tx=PRODUCTION artix7=PRODUCTION)
OPTION RUN_NGCBUILD = FALSE
OPTION STYLE = HDL
```

图 13：修改 MPD 文件

我们回到 XPS 界面下，点击菜单栏里的 PROJECT 按钮，点击 RESCAN USER REPOSITORIES。

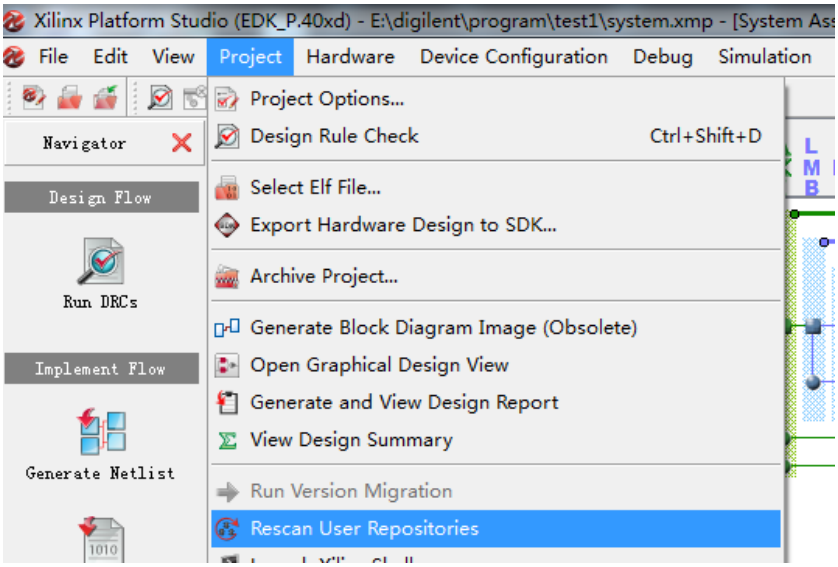


图 14：重新扫描文件

我们在工程界面选择 **PROJECT LOCAL PCORES**，双击添加 **PS2IP** 核。

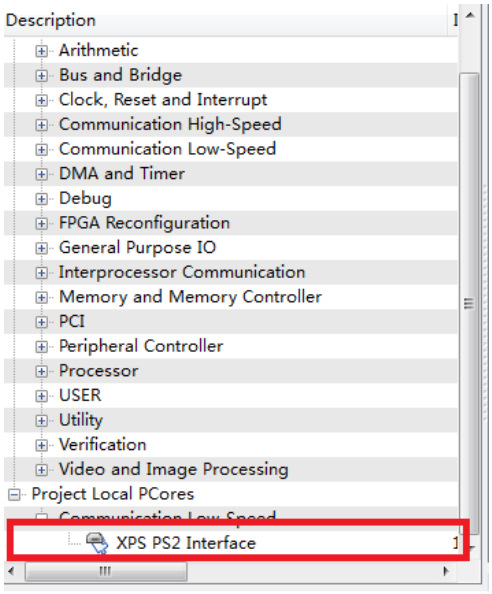


图 15：添加 PS2IP 核

2-1-7. 我们要配置刚刚添加的 **IP** 核，首先在 **BUS INTERFACES** 界面，点击 **BRIDGE** 的 **IP** 核，将它分别连接到 **AXI** 和 **PLB** 总线上。

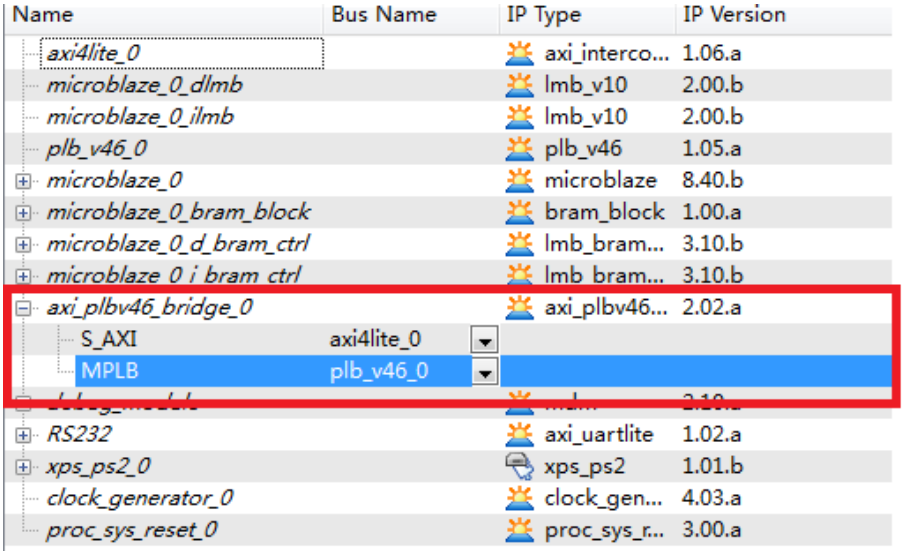


图 16：配置 BRIDGE IP

之后我们配置 **PS2IP** 核，将其连接到 **PLB** 总线上。

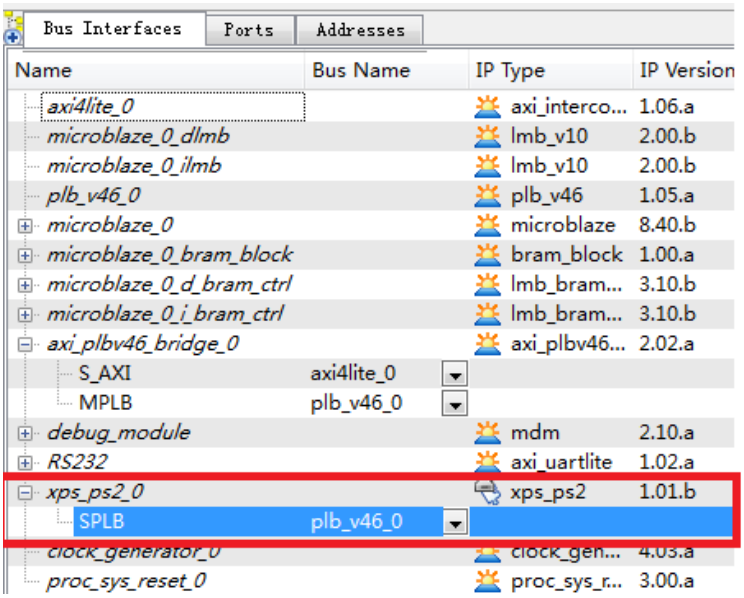


图 17：配置 PS2 IP

2-1-8. 我们开始配置端口，切换到 **PORTS** 界面,首先将鼠标放到界面最右边，出现 **port filter** 界面。勾选上 **Defaults.**

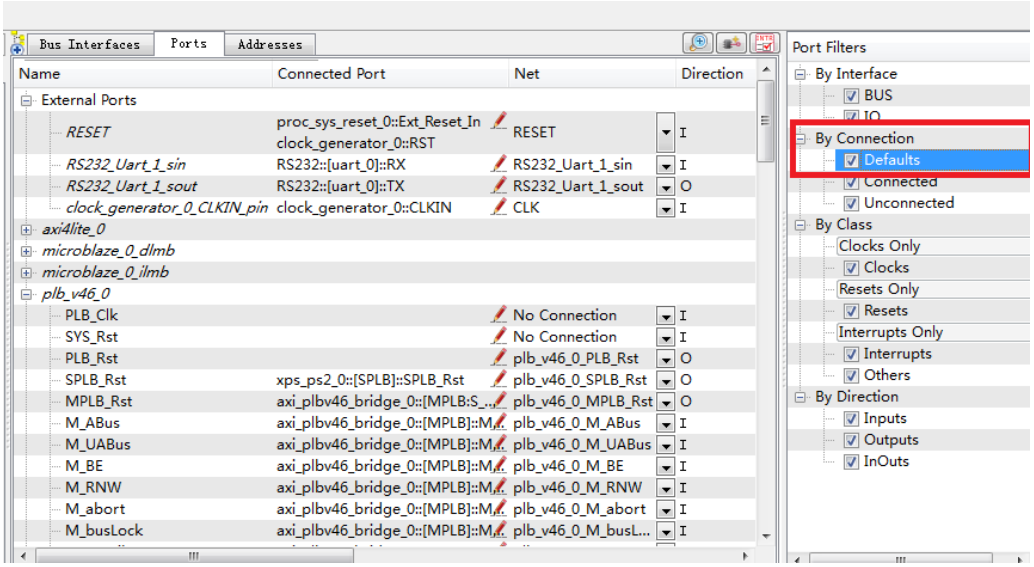


图 18:勾选 DEFAULTS

2-1-9. 下面我们连接 PLB 总线 IP 核的端口，将连接 PLB 时钟端到系统时钟，之后将复位端连接到 microblaze 的复位端。

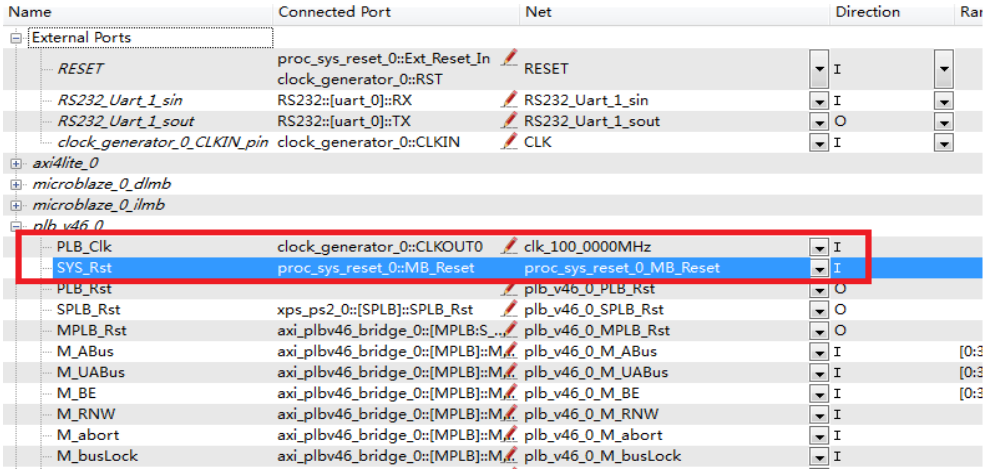


图 19：连接 PLB 时钟端和复位端

2-1-10. 下面配置 PS2 IP 核的端口，点开（IO_IF）PS2_0，将 DATA 和 CLK 端口设为外部端口。

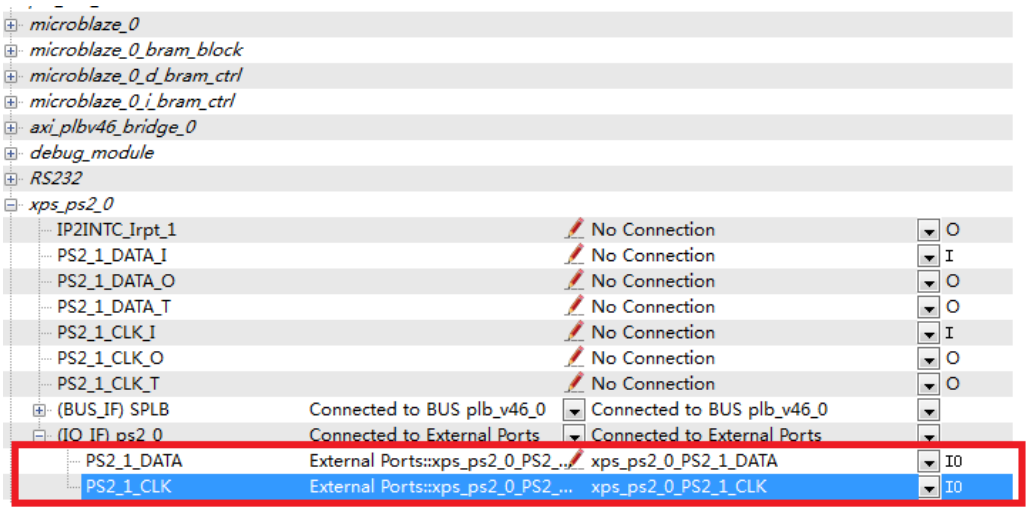
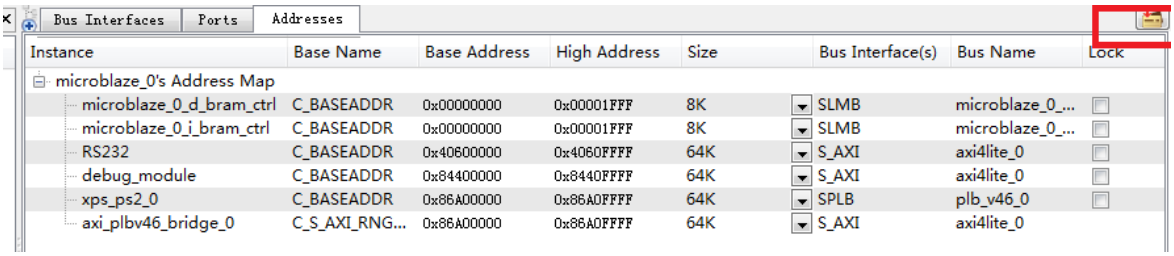


图 20：连接 PS2 端口

2-1-11. 下面开始配置地址，选择 address 按钮，点击 generate address。



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	Net	Direction	Range
External Ports				
RESET	proc_sys_reset_0::Ext_Reset_In clock_generator_0::RST	RESET	I	
RS232_Uart_1_sin	RS232::[uart_0]::RX	RS232_Uart_1_sin	I	
RS232_Uart_1_sout	RS232::[uart_0]::TX	RS232_Uart_1_sout	O	
clock_generator_0_CLKIN_pin	clock_generator_0::CLKIN	CLK	I	
xps_ps2_0_PS2_1_CLK	xps_ps2_0::[ps2_0]::PS2_1_CLK	xps_ps2_0_PS2_1_CLK	IO	
xps_ps2_0_PS2_1_DATA	xps_ps2_0::[ps2_0]::PS2_1_DATA	xps_ps2_0_PS2_1_DATA	IO	

图 21：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\system.ucf**，ucf 文件在右侧打开

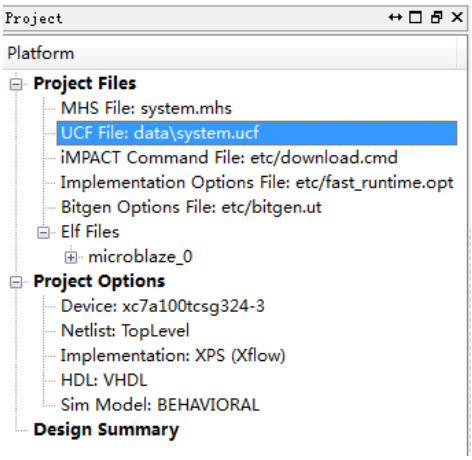
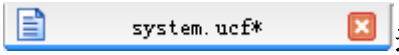


图 22：UCF 文件的位置

3-1-2. 这里我们手动输入 LOC（引脚位置）约束代码，如**图 23。点击保存。**

小技巧：可以通过 ctrl+s 的快捷键组合进行保存。如果 ucf 窗口下方的文件名旁边有*，如图所示：



这说明 ucf 文件经过编辑但还未保存。

```
1  ## This file is a general .ucf for the Nexys4 rev B board
2  ## To use it in a project:
3  ## - uncomment the lines corresponding to used pins
4  ## - rename the used signals according to the project
5
6  # Clock signal
7  NET "clock_generator_0_CLKIN_pin"   LOC = "E3"  | IOSTANDARD = "LVCMOS33";            #B
8  NET "clock_generator_0_CLKIN_pin" TNM_NET = sys_clk_pin;
9  TIMESPEC TS_sys_clk_pin = PERIOD sys_clk_pin 100 MHz HIGH 50%;
10
11 ## Switches
12 NET "RESET"       LOC = "U9"  | IOSTANDARD = "LVCMOS33";    #Bank = 34, Pin name = IO_L21
13
14
15 ## USB-RS232 Interface
16 NET "RS232_Uart_1_sin"          LOC = "C4"  | IOSTANDARD = "LVCMOS33";    #Bank = 35, Pi
17 NET "RS232_Uart_1_sout"         LOC = "D4"  | IOSTANDARD = "LVCMOS33";    #Bank = 35, Pi
18 #NET "RsCts"          LOC = "D3"  | IOSTANDARD = "LVCMOS33";    #Bank = 35, Pin name = IO_
19 #NET "RsRts"          LOC = "E5"  | IOSTANDARD = "LVCMOS33";    #Bank = 35, Pin name = IO_
20
21 ## USB HID (PS/2)
22 NET "xps_ps2_0_PS2_1_CLK"        LOC = "F4"  | PULLUP | IOSTANDARD = "LVCMOS33";    #Bank
23 NET "xps_ps2_0_PS2_1_DATA"       LOC = "B2"  | PULLUP | IOSTANDARD = "LVCMOS33";    #Bank
24
```

图 23：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**。

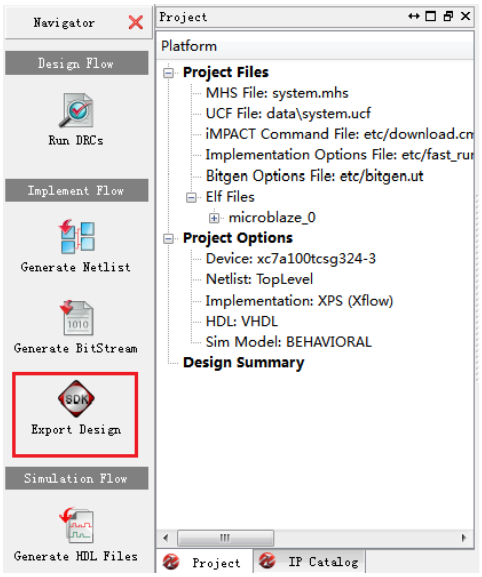


图 24：export design

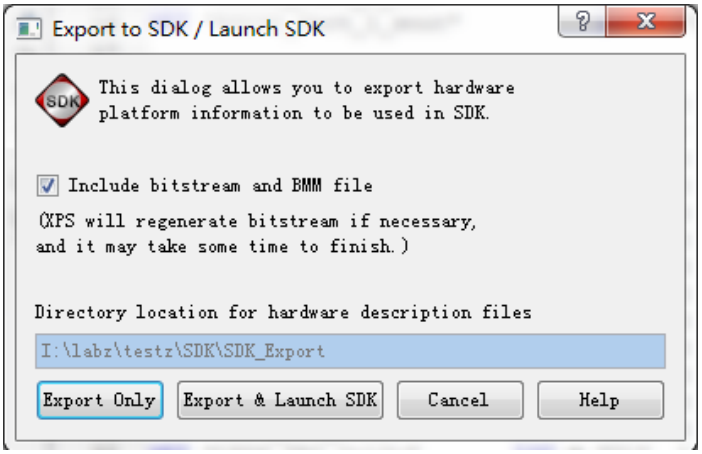


图 25：在弹出的对话框中选择 Export & launch sdk

单击 **Export&Launch SDK** 启动 **SDK** 软件。

3-1-4. 选择 SDK 导入路径

注意要具体到..\sdk\sdk_export

点击 ok

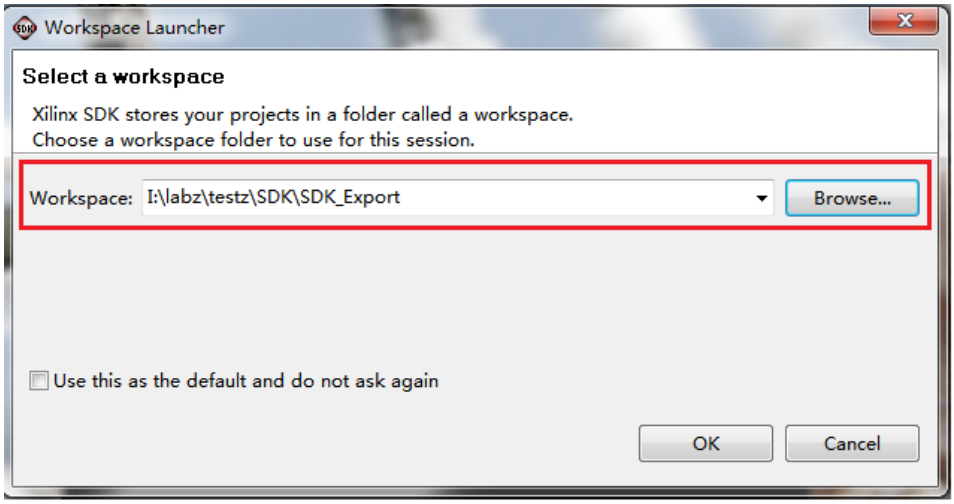


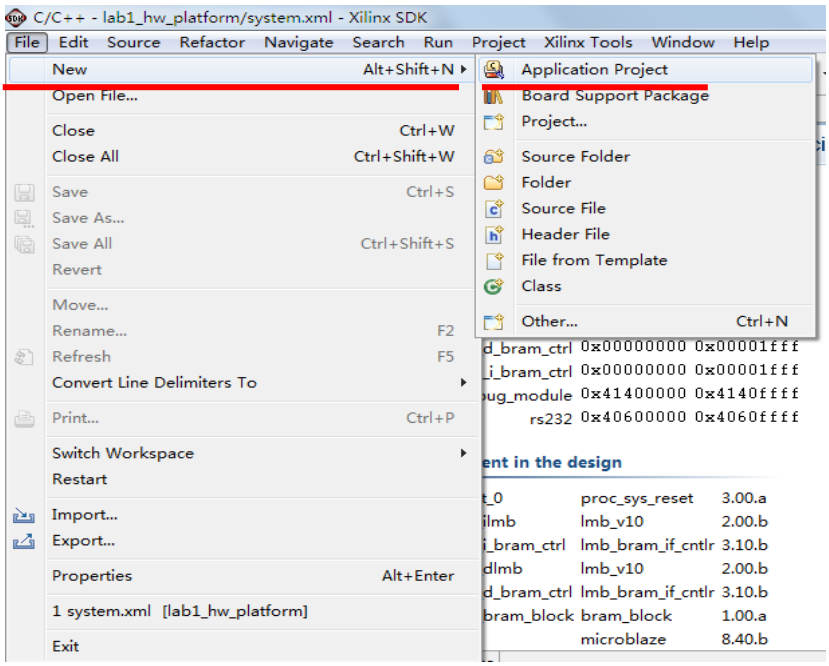
图 26：选择 SDK 导入路径

选择好本工程的路径之后单击 OK 启动 SDK。

第四步 添加 app

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**



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

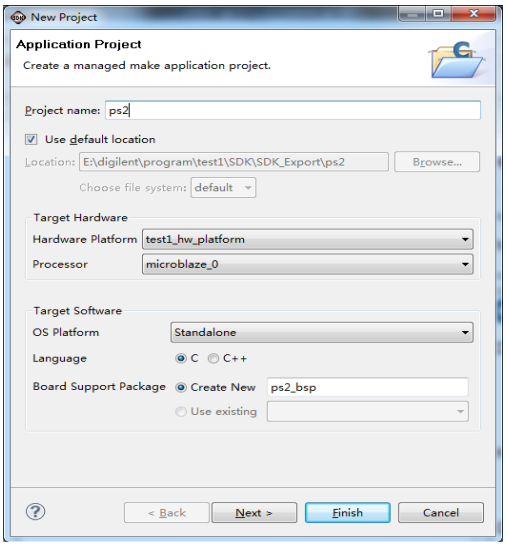


图 27：建立软件工程

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

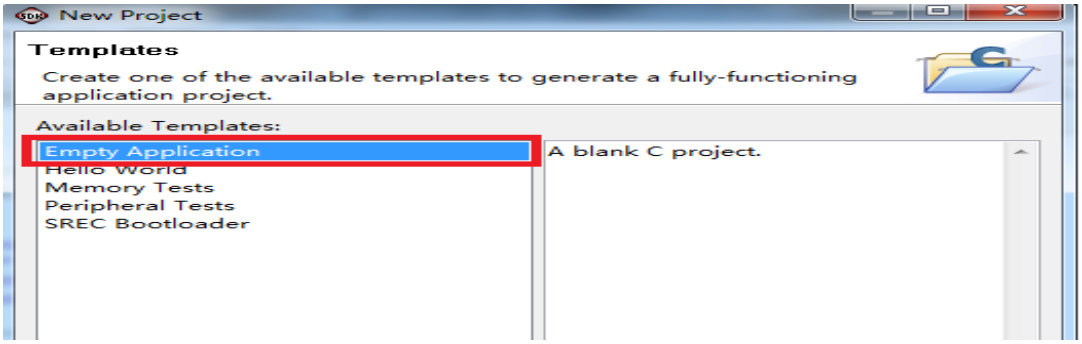


图 28：选择软件工程的模板

4-1-4. 因为开始时建立的是空工程，所以需要手动添加源文件，并输入文件名（注意后缀），如图 **29** 和 **30.**

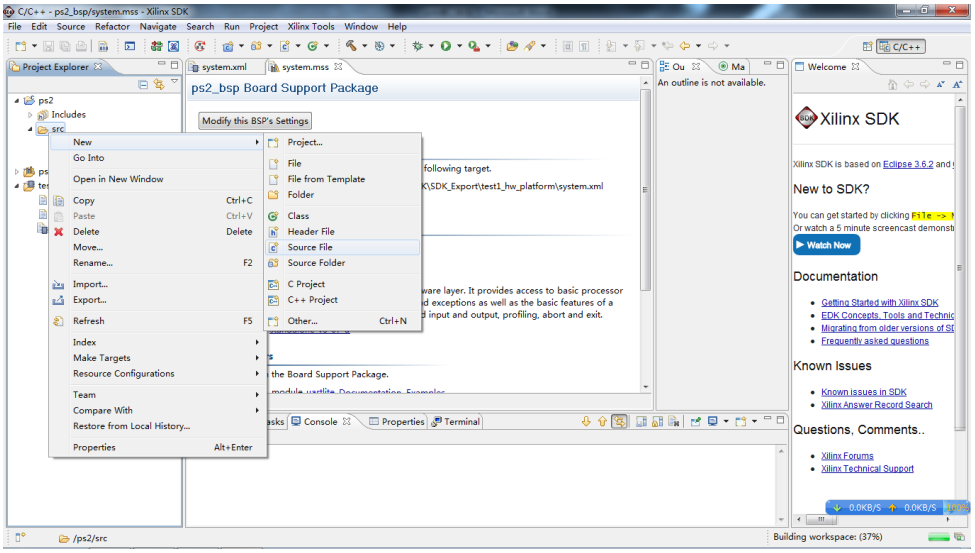


图 29：添加源文件

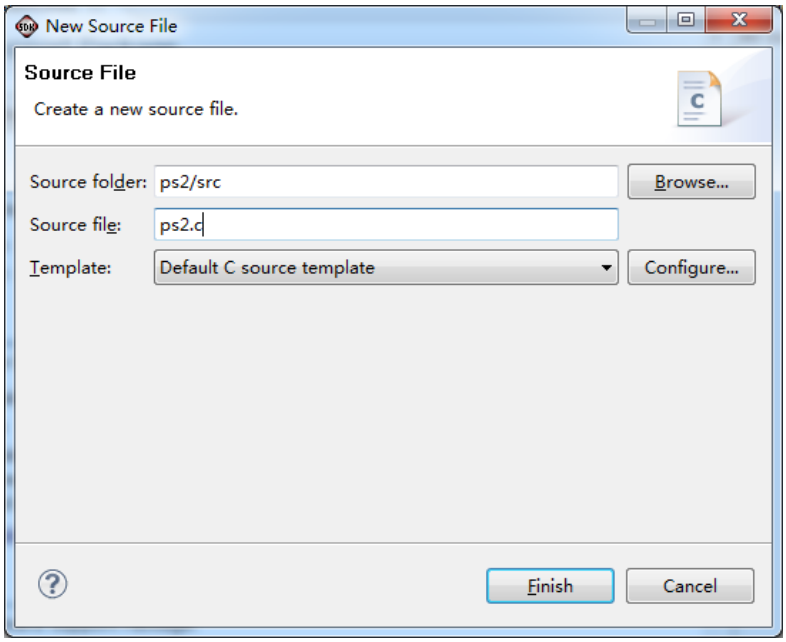


图 30：输入文件名

4-1-5. 添加完毕以后将 **SOURCE** 文件夹的 **USB.C** 文件打开将代码复制过来，查看这段代码：

```
/*
 * ps2.c
 *
 *  Created on: 2013-11-27
 *      Author: tp
 */


#include "xps2.h"
#include "xparameters.h"
#include "xstatus.h"
#include "stdio.h"

/************************** Constant Definitions *****************************/

/*
 * The following constants map to the XPAR parameters created in the
 * xparameters.h file. They are defined here such that a user can easily
 * change all the needed parameters in one place.
 */
#define PS2_DEVICE_ID       XPAR_PS2_0_DEVICE_ID


/**************************** Type Definitions *******************************/

/***************** Macros (Inline Functions) Definitions *********************/
```

```
#define TOTAL_TEST_BYTES    18  /* Total Number of bytes to be
                                transmitted/received */
#define KEYBOARD_ACK        0xFA    /* ACK from keyboard */
#define printf xil_printf       /* A smaller footprint printf */

/************************** Function Prototypes ******************************/

int Ps2PolledExample(u16 Ps2DeviceId);


/************************** Variable Definitions *****************************/

static XPs2 Ps2Inst;        /* Ps2 driver instance */

/*
 * Transmit Buffer contains data for glowing the LEDs on the
 * PS/2 keyboard.
 */
u8 TxBuffer[TOTAL_TEST_BYTES] = {0xED, 0x00, 0xED, 0x01,
                    0xED, 0x02, 0xED, 0x04,
                    0xED, 0x07, 0xED, 0x06,
                    0xED, 0x01, 0xED, 0x00,
                    0xED, 0x07};
/*
 * Receive Buffer.
 */
```

```
 * Receive Buffer.
 */
u8 RxBuffer;


/*****************************************************************************/
/**
*
* Main function that invokes the PS/2 polled example.
*
* @param    None.
*
* @return
*       - XST_SUCCESS if the example has completed successfully.
*       - XST_FAILURE if the example has failed.
*
* @note     None.
*
******************************************************************************/
int main(void)
{

    int Status;

    /*
```

```
     */
    Status = Ps2PolledExample(PS2_DEVICE_ID);
    if (Status != XST_SUCCESS) {
        return XST_FAILURE;
    }
    return XST_SUCCESS;

}
```

```
int Ps2PolledExample(u16 Ps2DeviceId)
{
    int Status;
    XPs2_Config *ConfigPtr;
    u32 Count;
    u32 BytesSent;
    u32 StatusReg;
    u32 BytesReceived;
    u32 Delay;


    /*
     * Initialize the Ps2 driver.
     */
    ConfigPtr = XPs2_LookupConfig(Ps2DeviceId);
    if (ConfigPtr == NULL) {
        return XST_FAILURE;
    }
    XPs2_CfgInitialize(&Ps2Inst, ConfigPtr, ConfigPtr->BaseAddress);
```

```
    Status = XPs2_SelfTest(&Ps2Inst);
    if (Status != XST_SUCCESS) {
        return XST_FAILURE;
    }

    printf("\r\nPS/2 Demo using Polled Mode\r\n");

    /*
     * Send the data to a PS/2 keyboard.
     */
    printf("Transmit some bytes to the PS/2 device \r\n");
    printf("Observe that the SCROLL/NUM/CAPS Lock Led's toggle \r\n");


    /*
     * Receive the data from a PS/2 keyboard.
     */
    Count = 1;
    printf("\r\n Press the Keys on the keyboard \r\n");
    printf("Echoing PS/2 scan codes from a PS/2 input device \r\n");
    while (1) {

        do {
            StatusReg = XPs2_GetStatus(&Ps2Inst);
        }while((StatusReg & XPS2_STATUS_RX_FULL) == 0);
        BytesReceived = XPs2_Recv(&Ps2Inst, &RxBuffer, 1);
```

```
        printf ("%x \r\n", RxBuffer);
        Count ++;
    }

    return XST_SUCCESS;
}
```

图 31：测试程序代码

可以根据自己的需要进行一些修改，修改后保存。

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 com4 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）

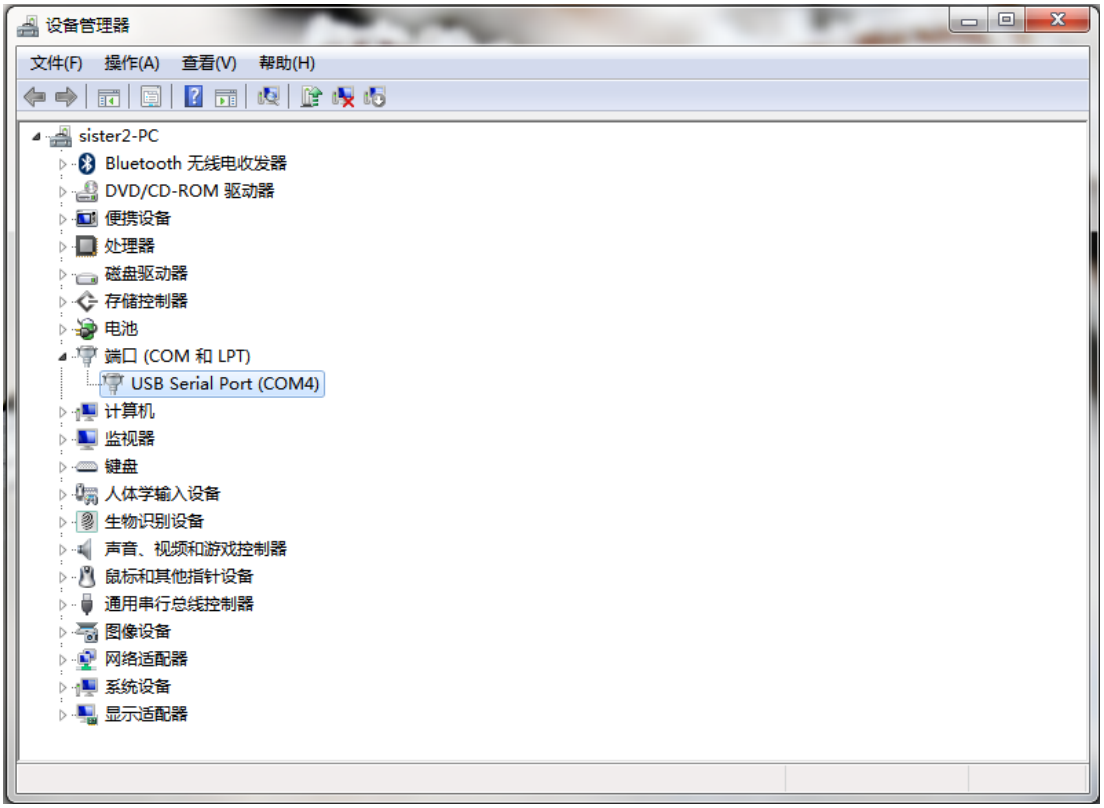


图 25：查看端口号

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。

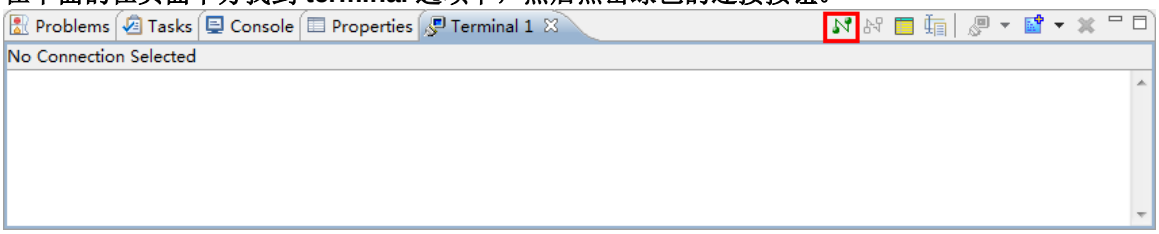


图 26：连接端口

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

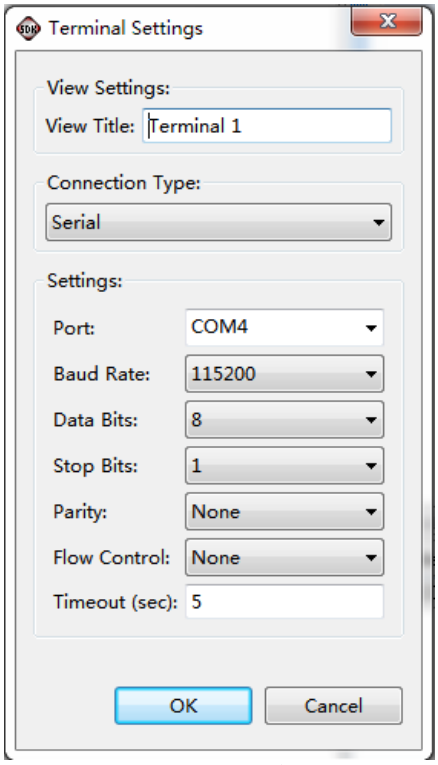
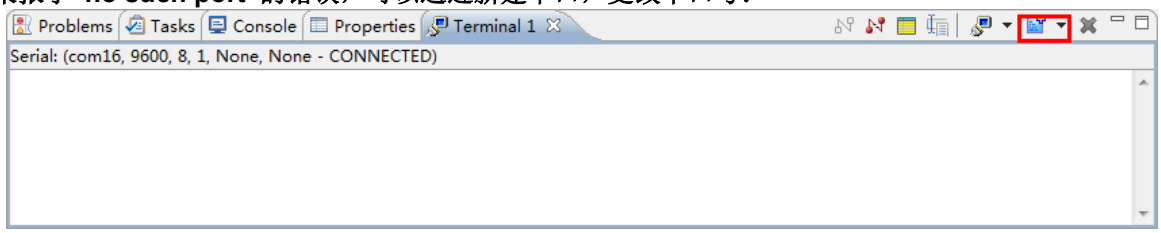


图 27：设置端口

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

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

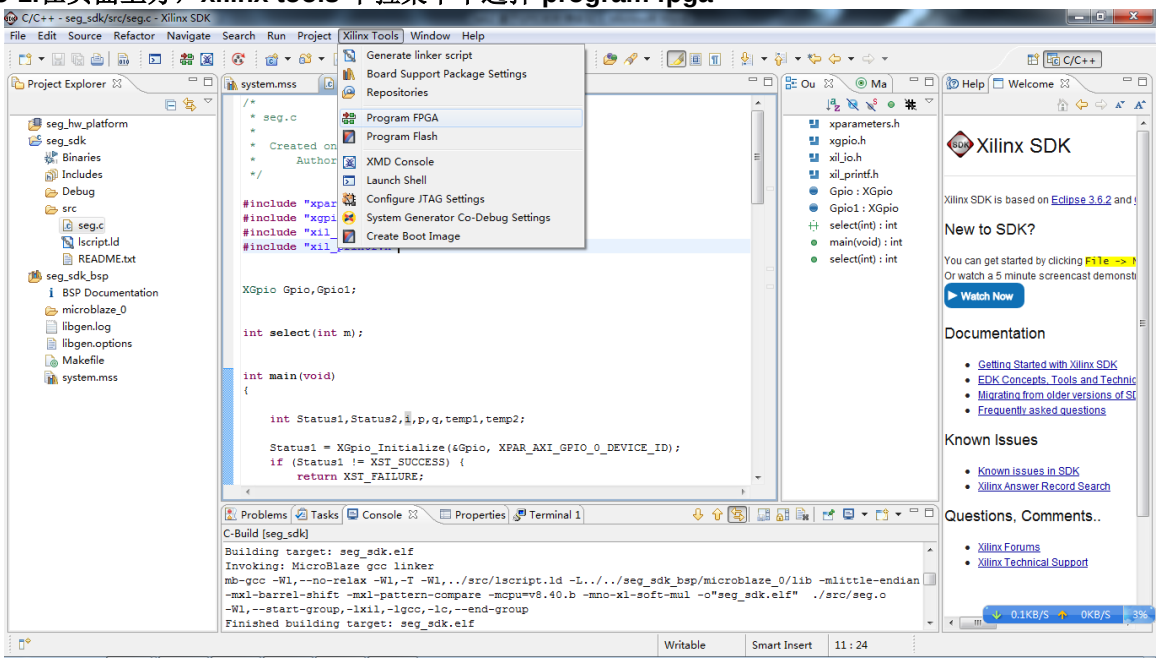


图 29：将 C 语言程序下载到开发板中

5-5-2.注意要选择正确的 elf 文件，并点击 program 运行程序：

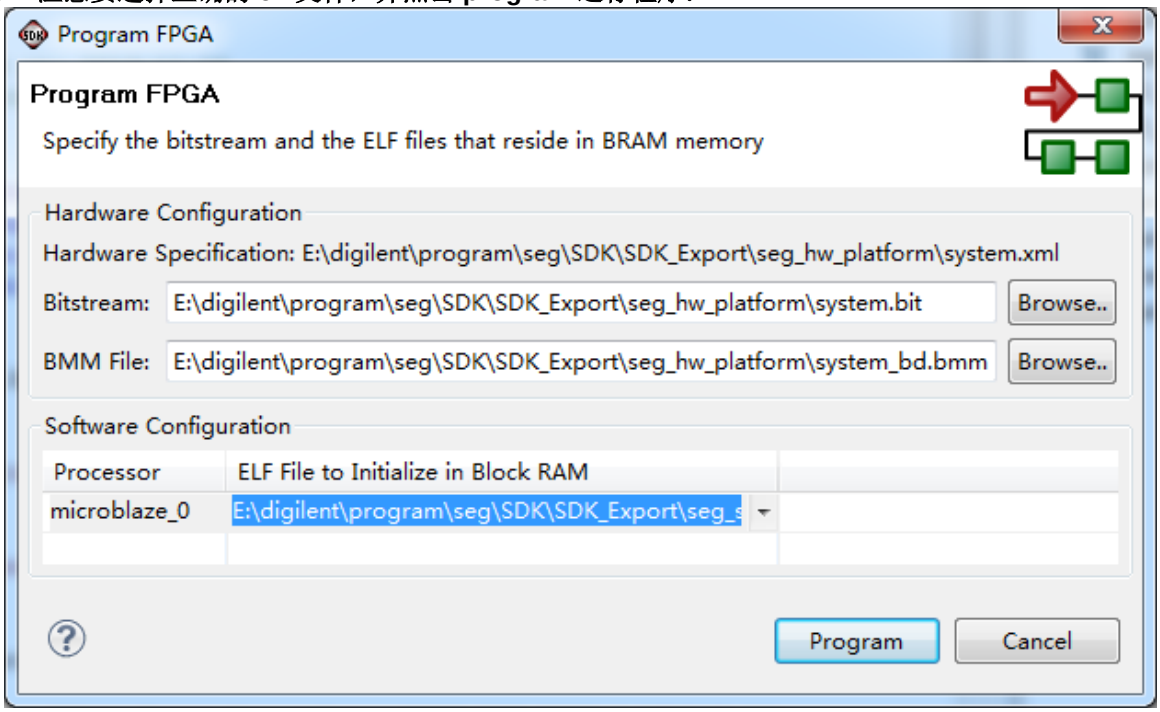


图 30：选择 elf 文件

5-5-3. 将程序下到板子上之后串口会出现如下现象

当键盘无输入的时候会出现如下现象

```
Serial: (COM10, 115200, 8, 1, None, None - CONNECTED)
PS/2 Demo using Polled Mode
Transmit some bytes to the PS/2 device
Observe that the SCROLL/NUM/CAPS Lock Led's toggle
```

当键盘上有输入的时候会出现如下现象（此图为按下键盘上的 K）

```
Serial: (COM4, 115200, 8, 1, None, None - CONNECTED)
Observe that the SCROLL/NUM/CAPS Lock Led's toggle

 Press the Keys on the keyboard
Echoing PS/2 scan codes from a PS/2 input device
AA
0
```

如果你想看到按键的通码就要按下键盘一个键 3、4 秒后松开（此图为 X 键），其中 22 是 X 键的通码。

```
Serial: (COM4, 115200, 8, 1, None, None - CLOSED)
AA
22
AA
0
22
22
```