Creating Basic Hardware and Software for the Virtex-5 Embedded Kit

This tutorial is an introduction to Embedded System development with the Virtex-5 Embedded Kit. In this tutorial you will design a custom embedded hardware system and develop simple application software without any OS/RTOS. For details on developing applications with Linux or other RTOSes, refer the Ecosystem Support document on the Virtex-5 Embedded kit page – http://www.xilinx.com/v5embedded.

Objectives

After completing this tutorial, you will be able to:

1. Create an Embedded system with the PowerPC processor for the Virtex-5 Embedded Kit (ML507 board) using Base System Builder (BSB) wizard
2. Generate and Download the FPGA design bitstream on the board using Xilinx Platform Studio (XPS)
3. Create a simple software application using Eclipse-based Platform Studio Software Development Kit (SDK)
4. Debug the software application using the Eclipse IDE (SDK)

Requirements

Embedded Development HW/SW Kit – Virtex-5 FX70T PowerPC and MicroBlaze Processor Edition

The Virtex-5 Embedded Kit includes all of the items listed below ***except the HyperTerminal program*** needed for RS232 UART output.

Hardware Requirements

- ML507 (Virtex-5 FX70T) board
- Xilinx Platform Cable USB
- RS232 cable

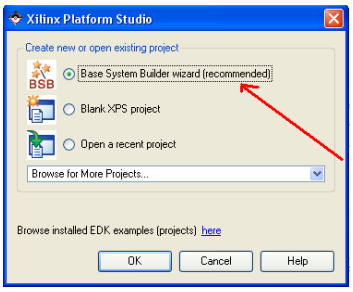
Software Requirements

- ISE Design Suite 10.1 (including EDK 10.1) with Service Pack 2
- HyperTerminal program connecting the PC and the FPGA board at 115,200 baud

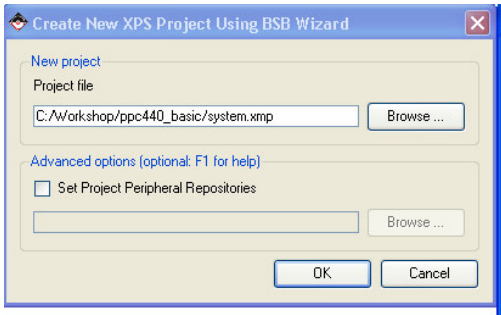
Step 1 – Create Custom Embedded System - Hardware

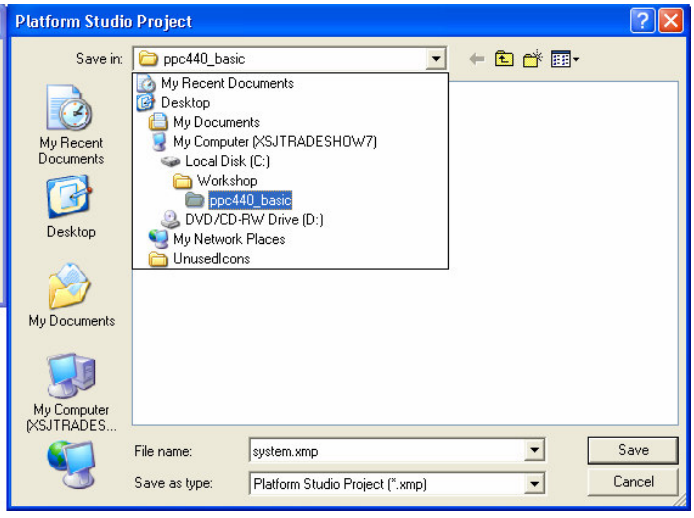
1. Open Xilinx Platform Studio

2. Create a new project:
   - Select **"Base System Builder wizard"** and Click **OK**



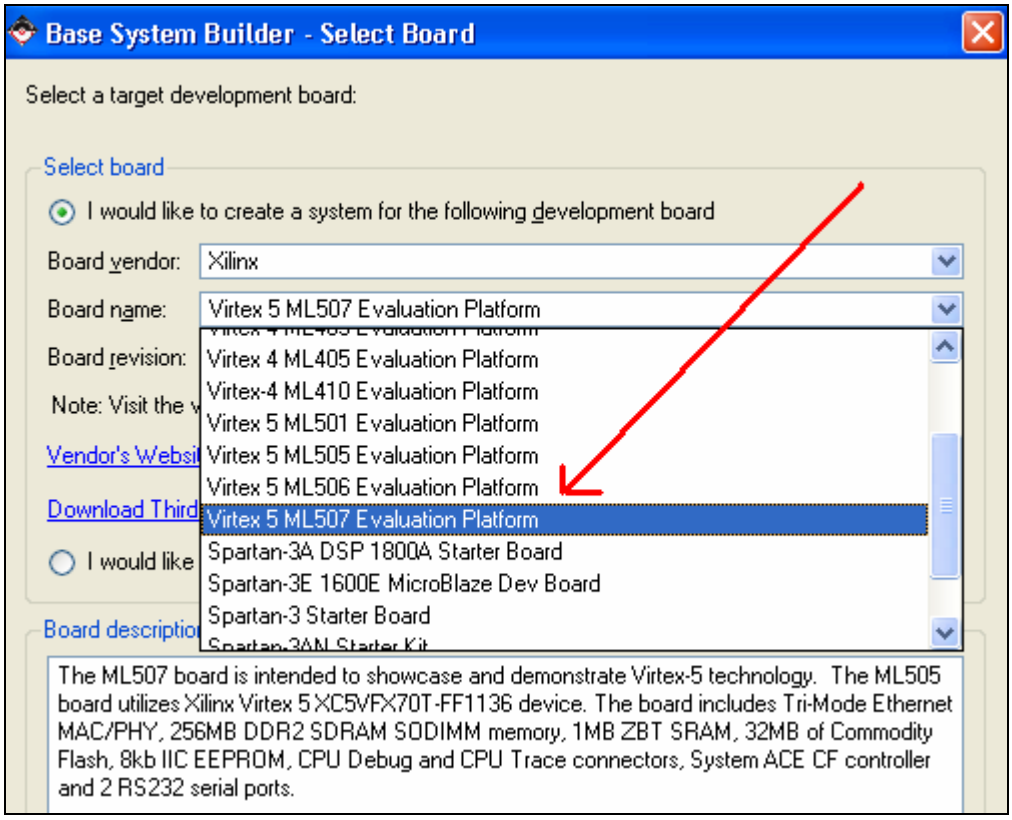
3. Select the design directory for new project:
   - Click the **Browse** button and browse to a location on your computer: **C:\Workshop\ppc440_basic**
   - The file name should be: **system**
   - Click **Save**
   - Click **OK**





4. Next the Wizard will give options on how to create the system:
   - Select "**I would like to create a new design**"
   - Click **"Next"**

5. Select the target development board:
   - Select Board vendor:"**Xilinx**"
   - Select Board name: **"Virtex-5 ML507 Evaluation Platform"**
   - Click **"Next"**



6. Select the processor:

With the BSB wizard you can design either MicroBlaze or PowerPC-based systems. For this tutorial, you are going to use the PowerPC 440 processor available in Virtex-5 FXT FPGAs.

- Select "**PowerPC**"
- Click **"Next"**

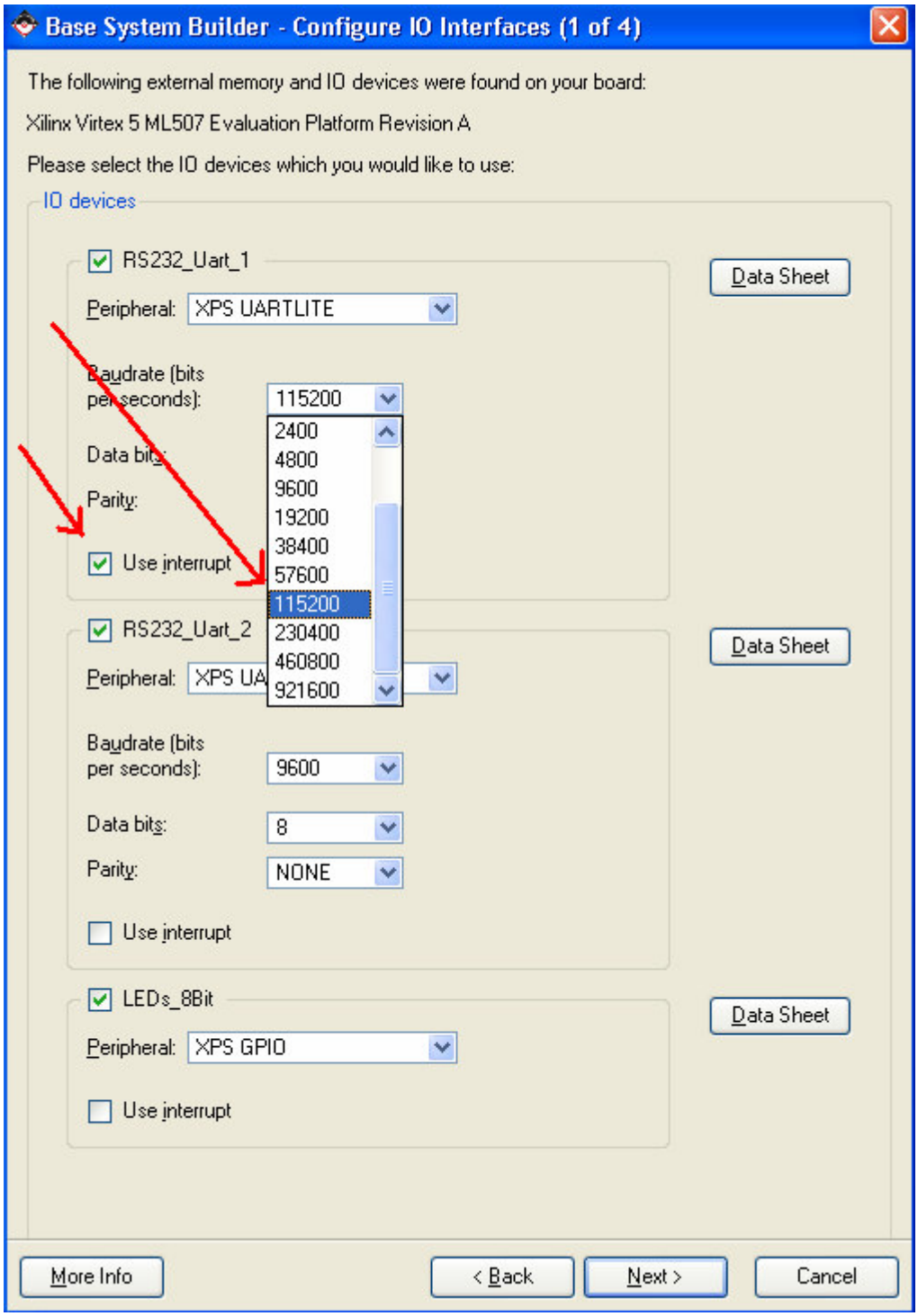
7. Configure the processor options:

You can modify options like Processor or Bus Frequency, FPU, etc. Make sure the "FPGA_JTAG" option is enabled for Debug setting. You will need this for Software debugging in the next section of this tutorial.

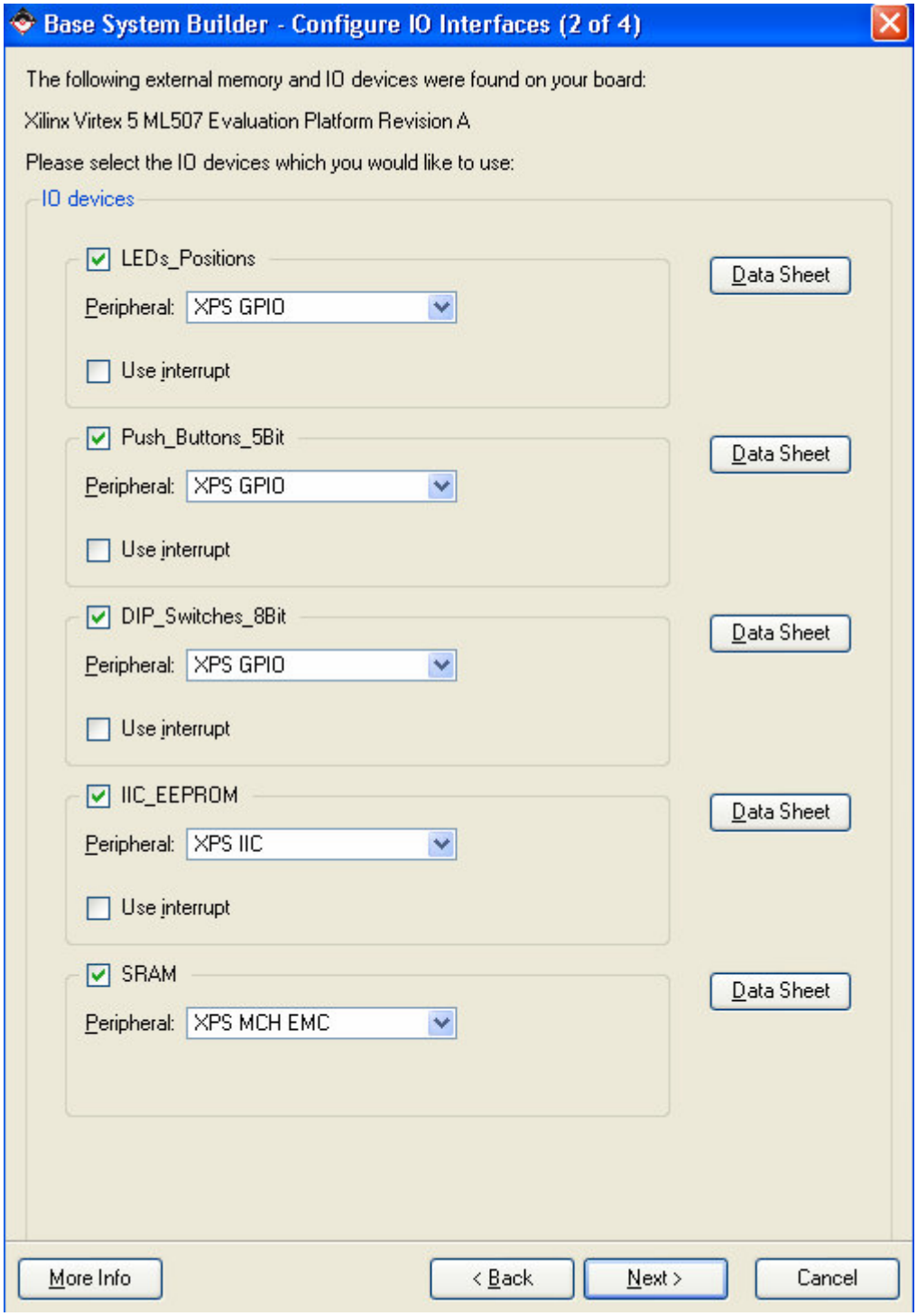
For now, leave the defaults as-is. You can modify the system later.

- Click **"Next"**

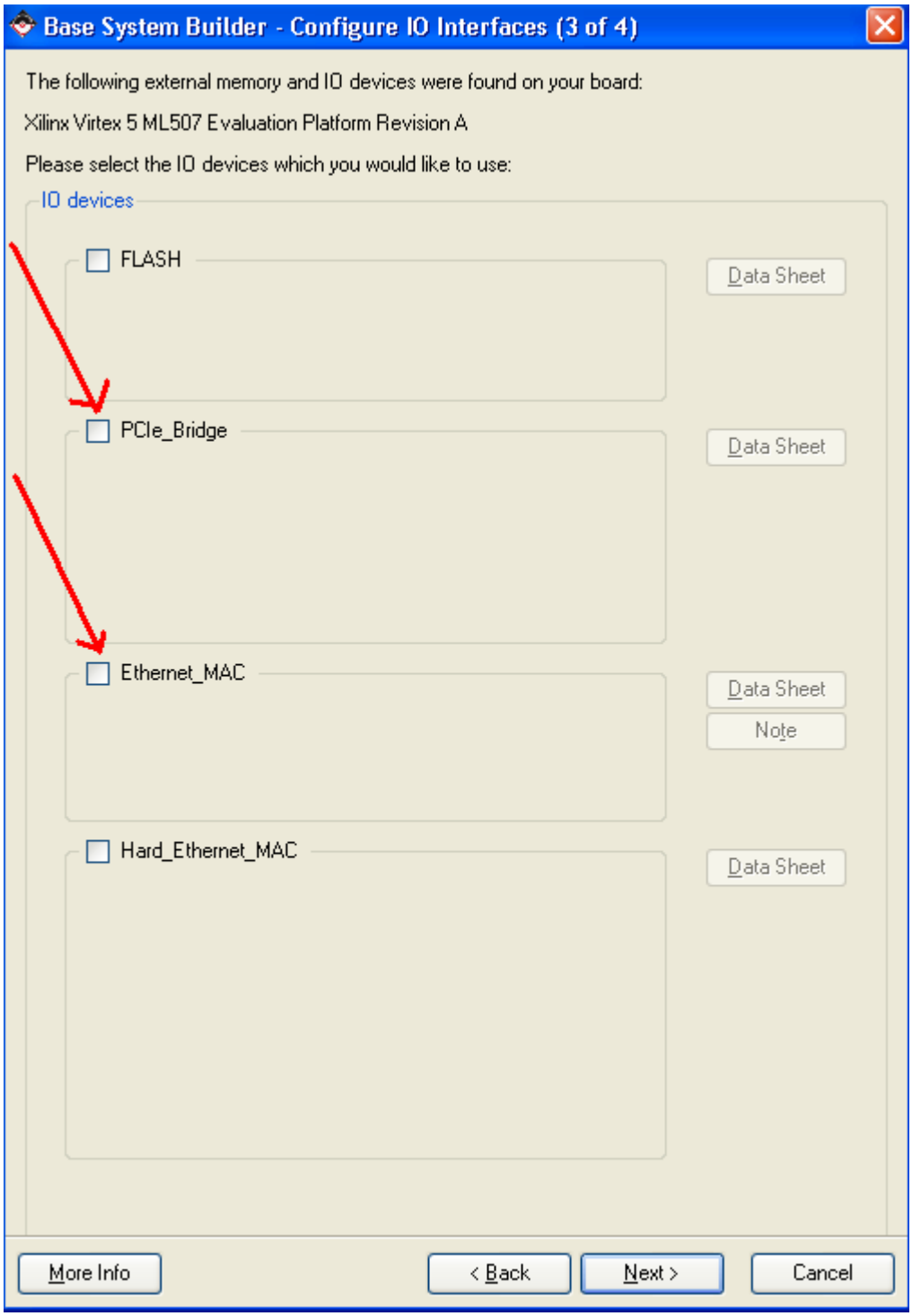
8. Configure the I/O interfaces of your embedded system. Here you can select the right set of Peripherals you need for your target system:
   - For "RS232Select the drop down for **"Baudrate"** and choose **"115200"**
   - Enable the check box **"Use Interrupt"**
   - Click **"Next"**



9. Continue configuring the I/O interfaces: (page 2 of 4). Leave the default options as-is.
    - Click **"Next"**

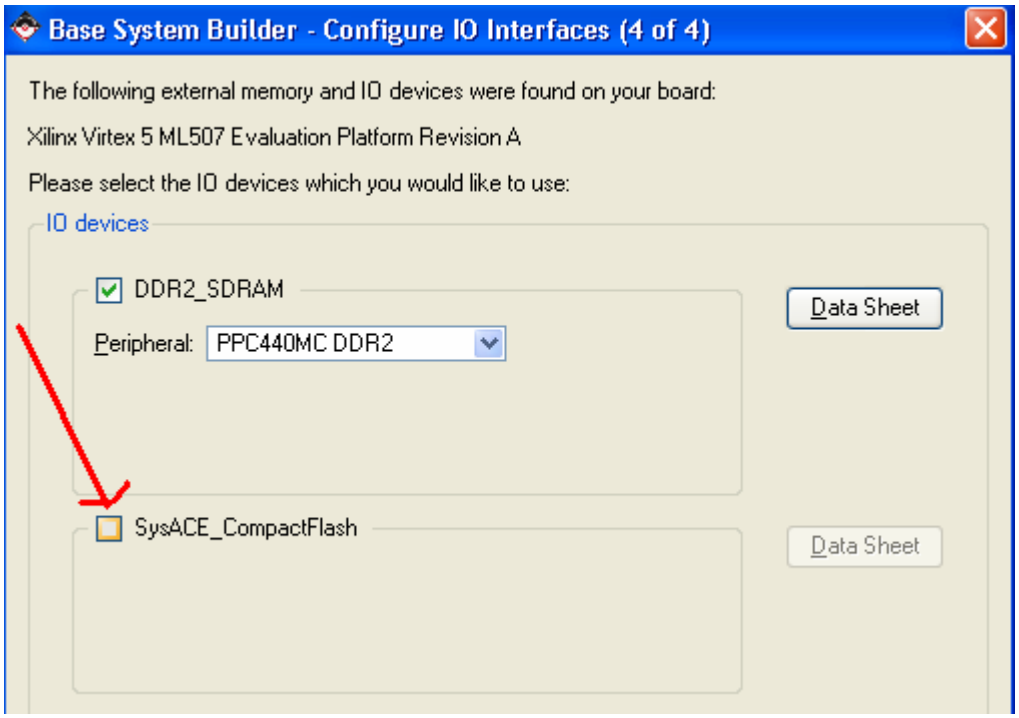


10. Continue configuring the I/O interfaces. (page 3 of 4) Unselect the PCI Express Bridge and Ethernet MAC as you will not be using it in this tutorial. This will make the design smaller and also save some time when you create the FPGA netlist/bitstream for this tutorial:
    - **Unselect "PCIe_Bridge"**
    - **Unselect the "Ethernet MAC"**



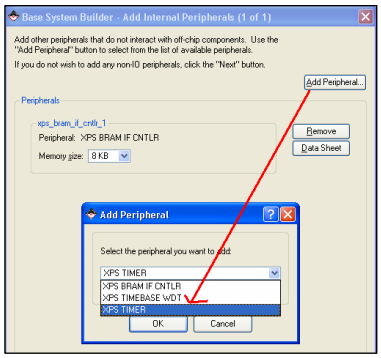
11. Continue configuring the I/O interfaces (page 4 of 4). Unselect the SystemACE peripheral as you will not be using it in this tutorial. This will make the design smaller and also save some time when you create the FPGA netlist/bitstream for this tutorial:

- **Unselect SysACE_CompactFlash**
- Click **"Next"**



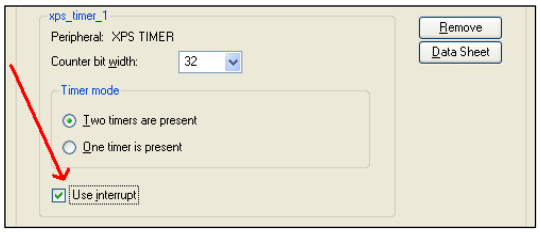
12. Add some internal on-chip peripherals:

- Click **"Add Peripheral"**
- Select **"XPS Timer"**
- Click **"Ok"**



13. Customize the Timer by enabling Interrupts:

- Click **"Use Interrupt"** for the timer peripheral
- Click **"Ok"**



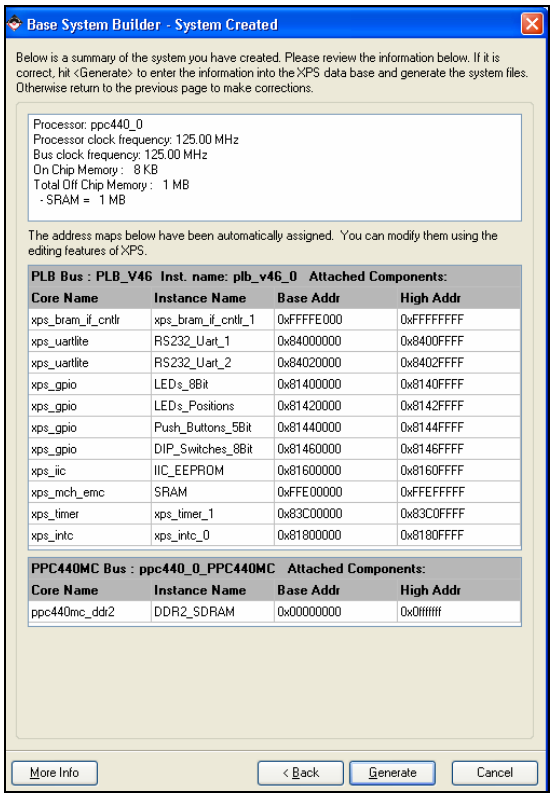
14. Next the wizard will help you create some sample programs that will do some diagnostics on your custom embedded system.

    You can leave the defaults and modify the software later:

    - Click **"Next"**
    - Click **"Next"**
    - Click **"Next"**

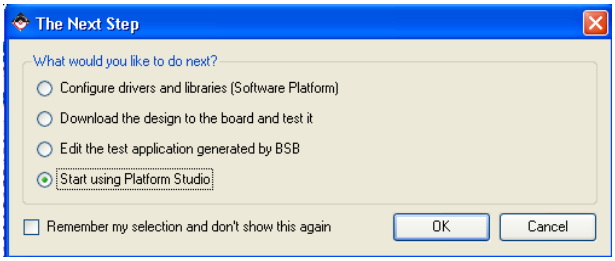
15. Now you have finished creating a basic system. The wizard will let you review your selections and generate the system:

    - Click **"Generate"**
    - Click **"Finish"**



16. If the following optional dialog appears, you can leave the defaults as-is.

    - Leave the default **"Start using Platform Studio"**
    - Click **"OK"**

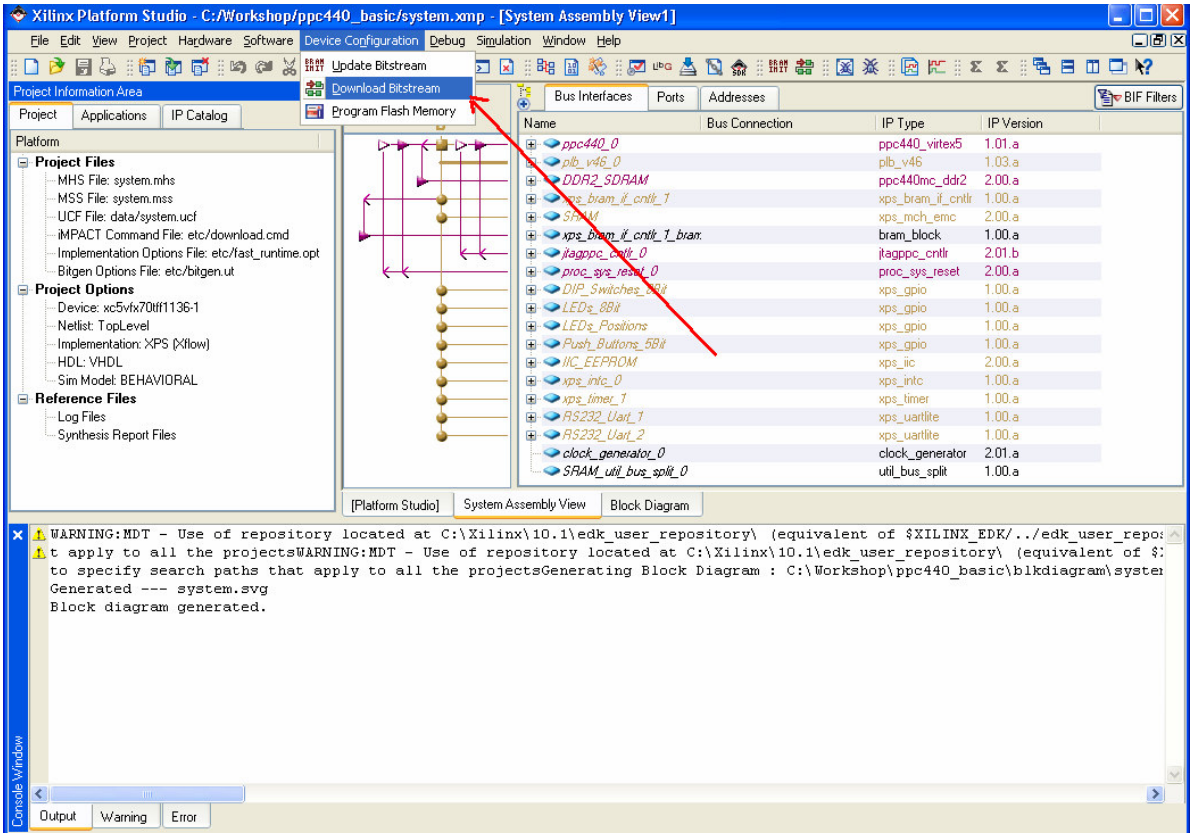


Step 2 – Build and Download the Hardware system

1. From the Xilinx Platform Studio (XPS) IDE, create the FPGA bitstream and download the hardware design onto the Spartan FPGA board as follows:

- Select the menu "**Device Configuration-> Download Bitstream**"

NOTE: This step will take almost 30 minutes. Do not close the window !



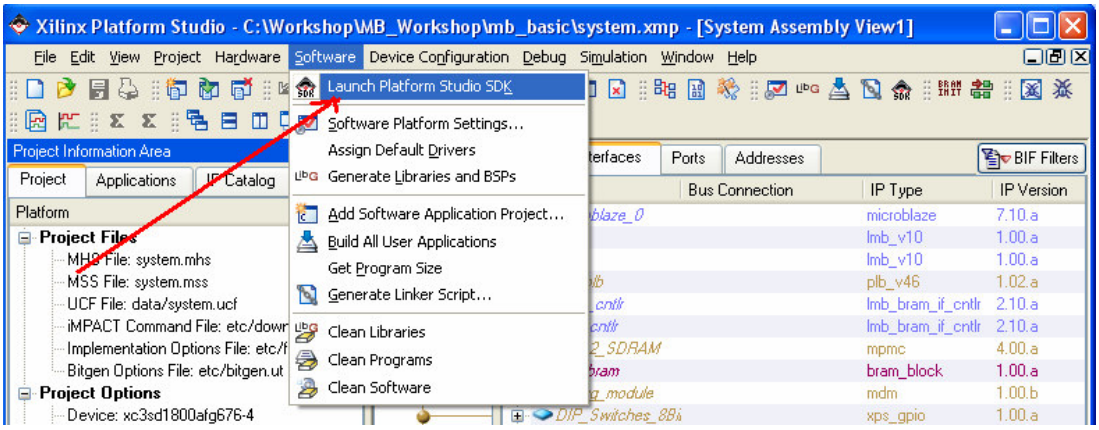
This will create the FPGA design files, synthesize the netlist and also place and route the design creating the bitstream. The FPGA on the board will also be configured with this new bitstream containing your custom embedded system.

Congratulations. Your Custom Hardware is now ready. Next you will write and program some software.

Step 3 – Develop Embedded software

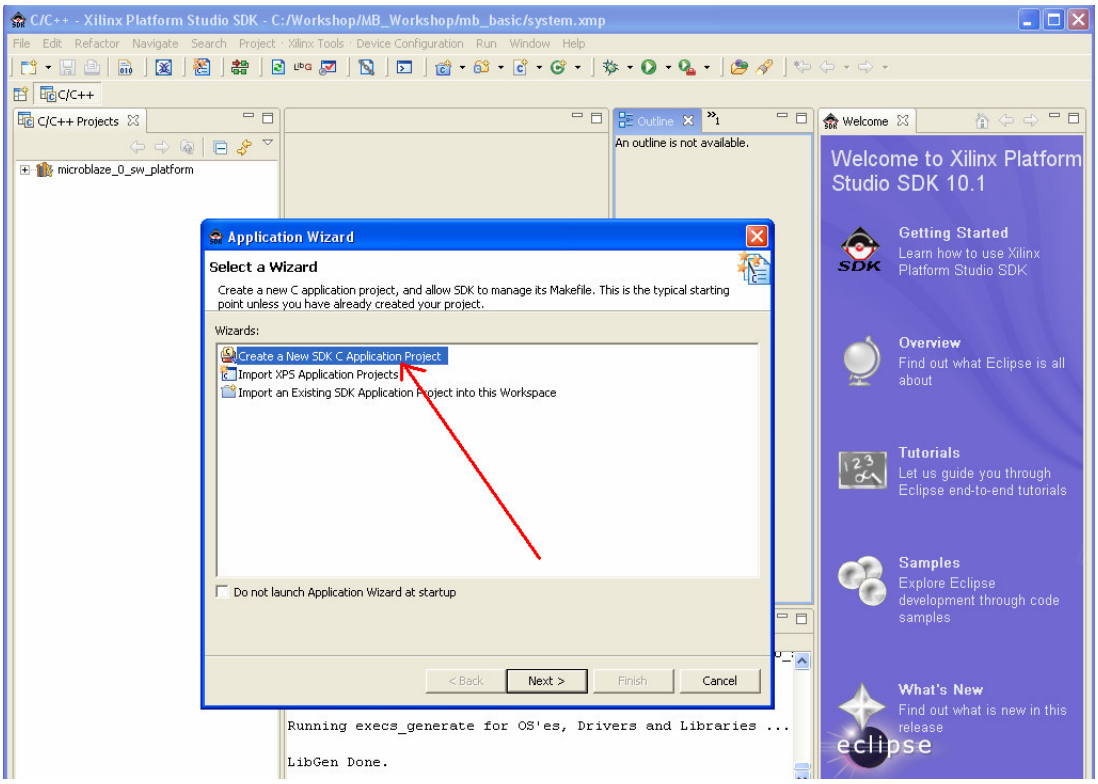
1. From the Platform Studio IDE, launch Eclipse IDE for software development:

- Select the menu **"Software->Launch Platform Studio SDK"**

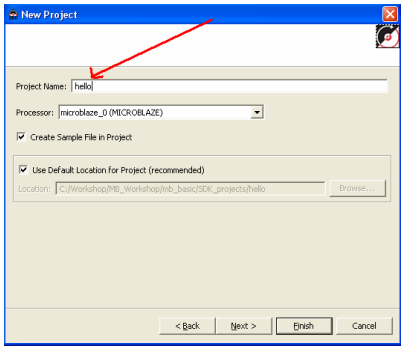


2. In SDK, start creating a new software application using the Application Wizard:

- Select **"Create a New SDK C Application Project"**
- Click **"Next"**

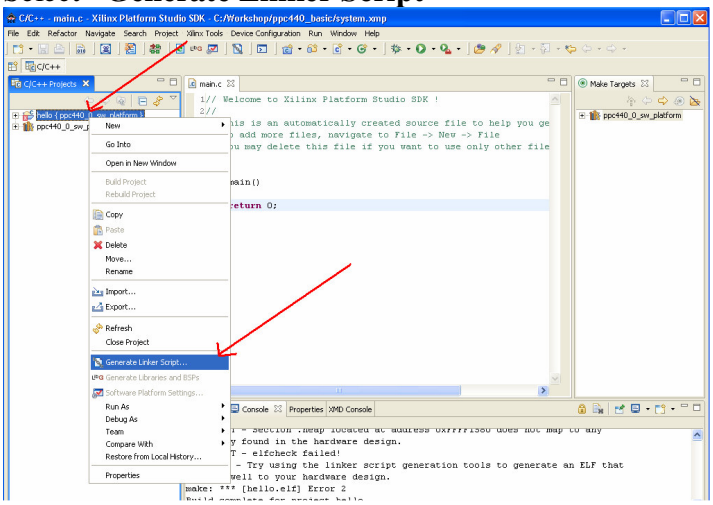


3. Provide a name to the new project
   - Type **"hello"**
   - Click **"Next"**

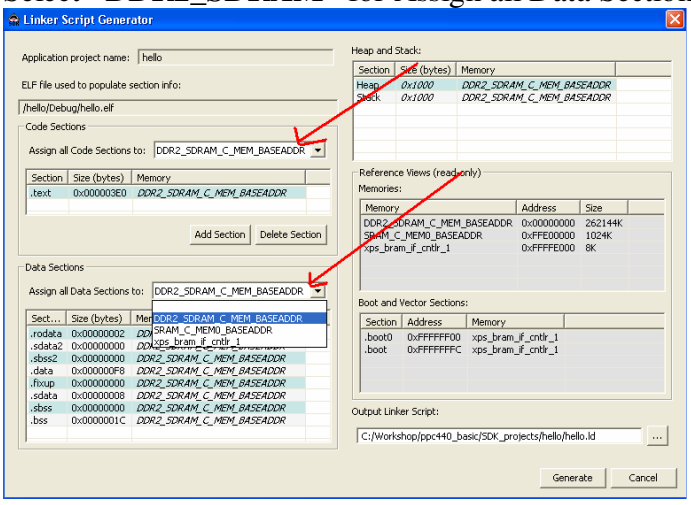


   Use the rest of the project defaults
   - Click **"Next"**
   - Click **"Finish"**

4. Now you have to edit the Linker Script of the program to make sure it will run properly in hardware from DDR2 external memory.
   - Right Click on the "**hello {ppc440_0_software_platform }**"
   - Select **"Generate Linker Script"**



   In the Linker Script Generator dialog
   - Select **"DDR2_SDRAM"** for Assign all Code Section to:
   - Select **"DDR2_SDRAM"** for Assign all Data Section to:



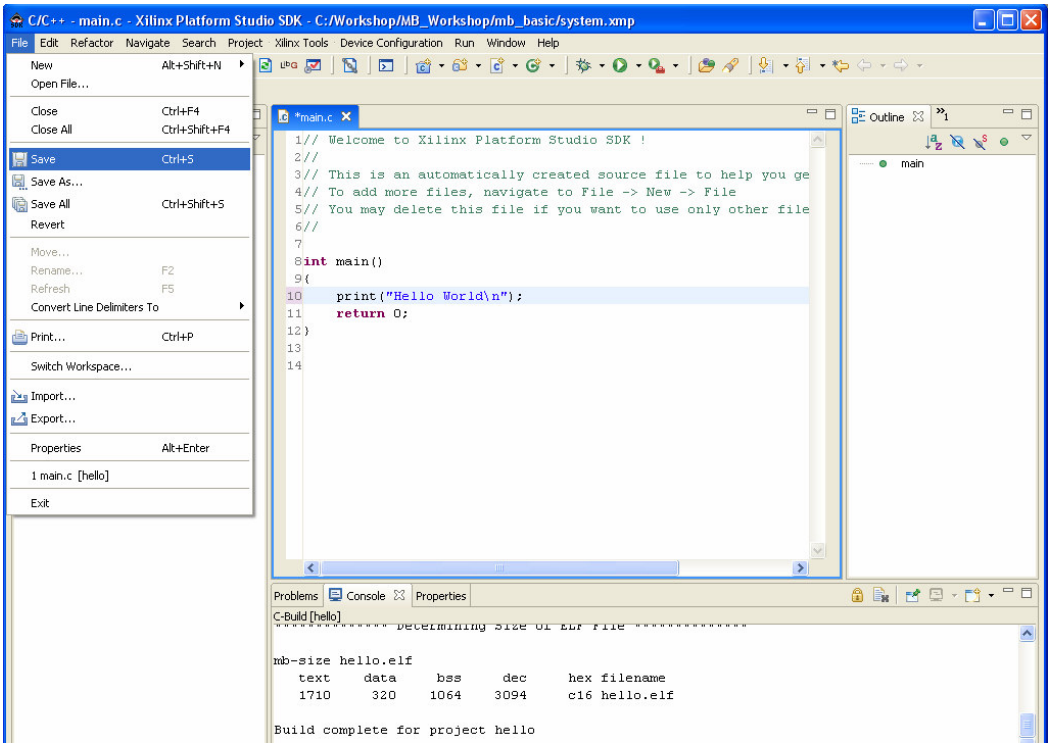
5. In the Editor view, edit the default program “main.c” and save it:

- Insert the following line after main() {

```
print("Hello World");
```

NOTE: It is print() and not printf().

- Select the menu **“File-> Save”**



Saving the program will also compile it and result in “hello.elf” as you see in the console view.

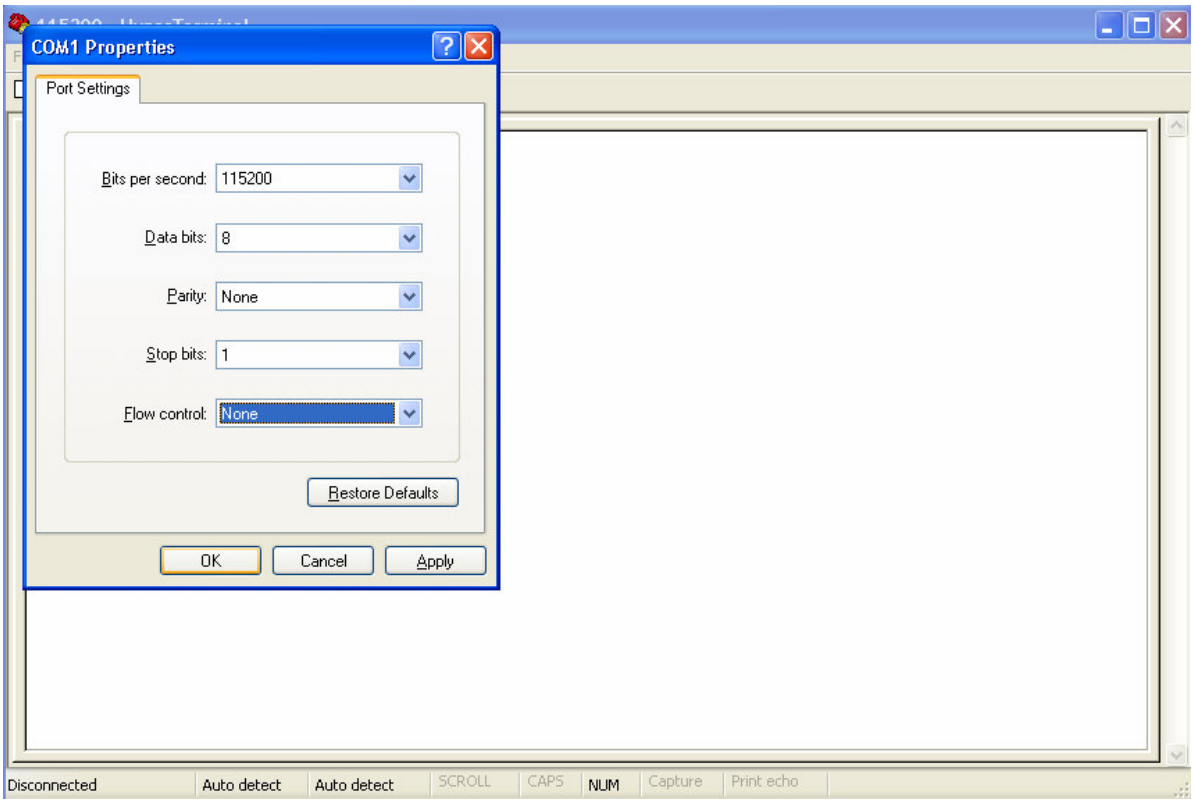
Now you are ready to download and debug the program on the board.

SETUP – Setting Up Hyperterminal

1. Before debugging the Hello world program, you have to open the RS-232 serial communication with the board using the Hyperterminal console program and set the Baud Rate to be 115200 baud.

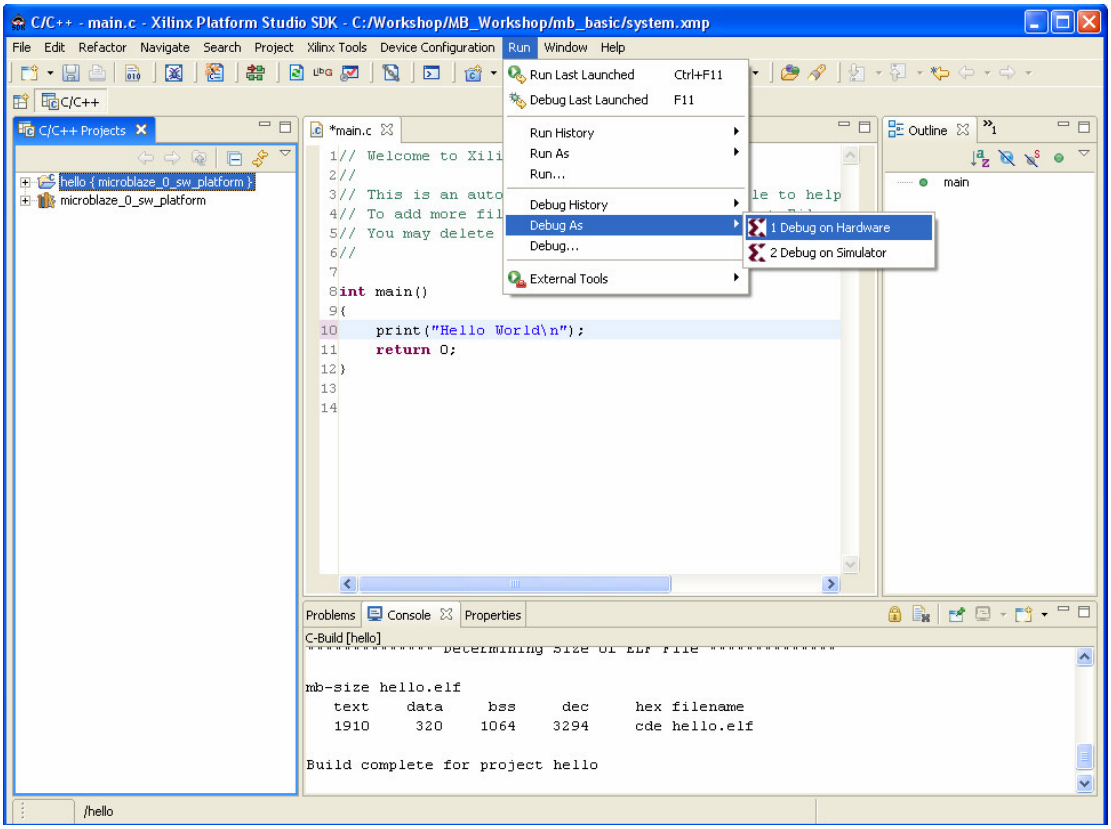
    - From your Windows Desktop, select **Start->All Programs->Accessories->Communications->Hyperterminal**
    - In Hyperterminal, give a name "**115200**" the Serial Connection
    - In Hyperterminal, **select "Connect Using" and select the Serial port on your PC (Usually it is Com1).**
    - In Hyperterminal, **Port Settings tab, select Bits per second as 115200, Data bits as 8, Parity as None, Stop Bits as 1 and Flow control as None. Then Click "Ok"**

The output of your Hello world program can be seen in this console.



Step 4 – Debugging software program

1. From SDK, start debugging the program as follows:
   - Select the Menu item **"Run->Debug As->Debug on Hardware"**

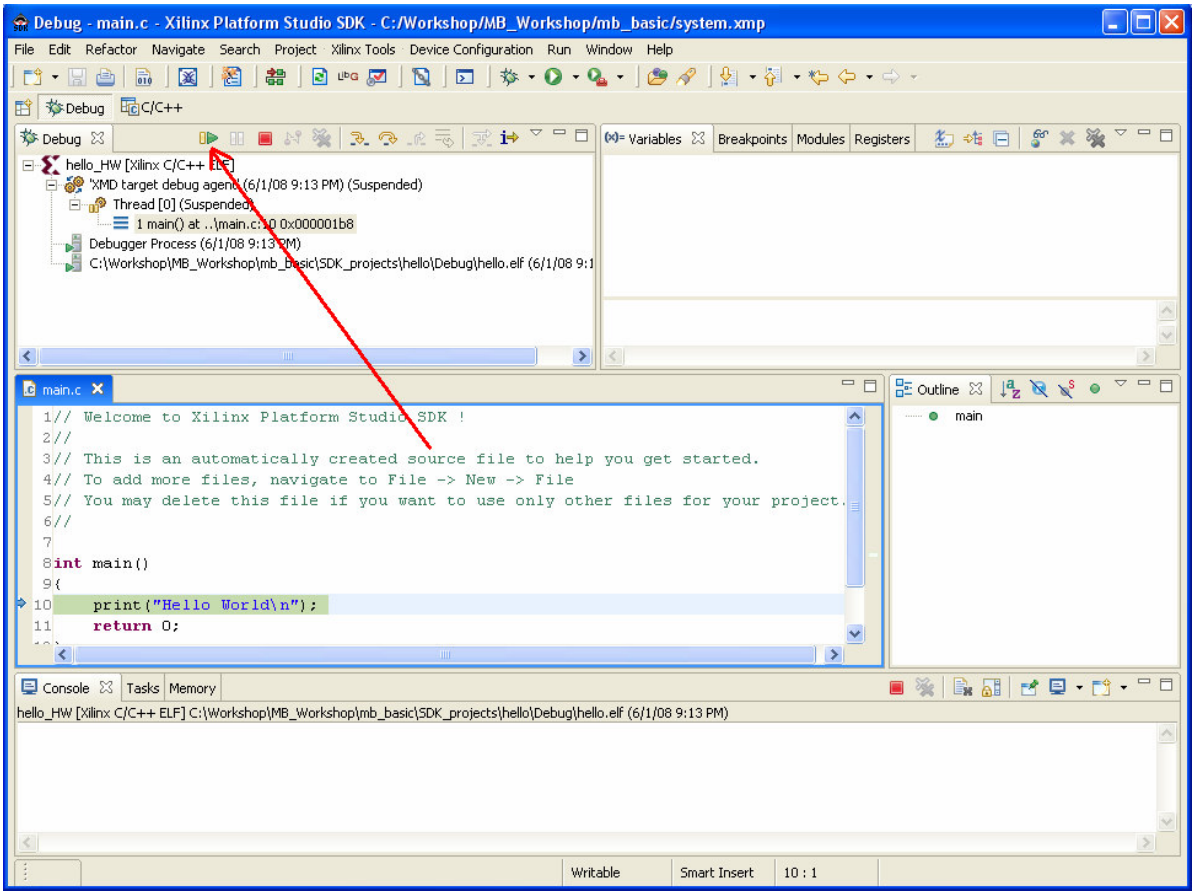


SDK will now change to the Debug Perspective. SDK will also connect to the FPGA board via the USB cable, connect to the MicroBlaze processor and download the program to memory.

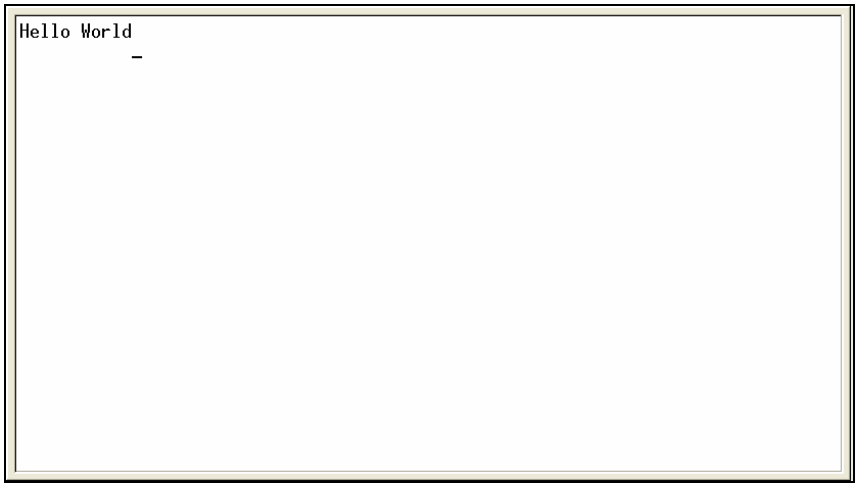
Now you are ready to debug the program using the Eclipse Debug perspective.

2. In the Debug perspective, click on the resume button (looks like ***Play*** button):
   - Click the "**Resume**" button

   *NOTE: You can alternatively select **"Run->Resume"***



You should now see the output of the program on the Serial console (Hyperterminal) as follows:



Congratulations. Your Software application is now running on your custom hardware platform.

Additional Exercises

You can now use the Eclipse IDE to modify your program and debug it further. Following are some other Software exercises you can do to import the board diagnostic programs created by Platform Studio and test the LEDs, Buttons and other peripherals on the board:

1. In the SDK IDE source code perspective,
    a. Select Xilinx Tools->Launch Application Wizard
    b. Import Test_App_Memory and Test_App_Peripheral programs.
    c. Compile and Run (or Debug) these programs

2. While debugging,
    a. View Registers, Memory
    b. Single Step through the program using Step, Next
    c. Toggle between assembly and source views

Inside Xilinx SDK IDE, Refer Help->Help Contents for documentation on additional SDK features.