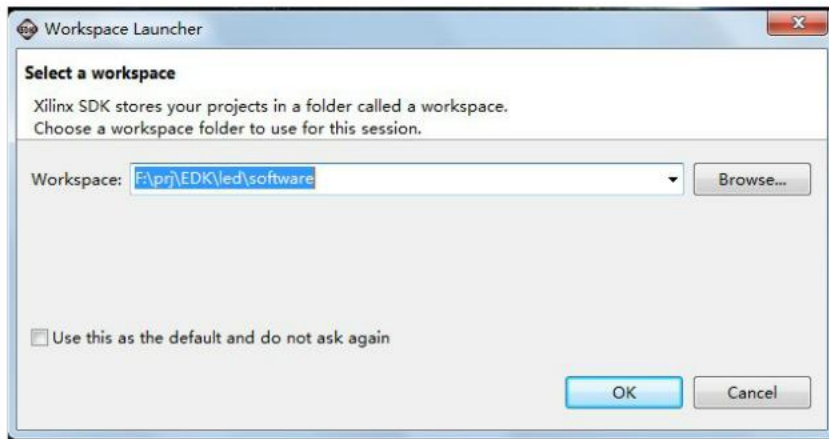


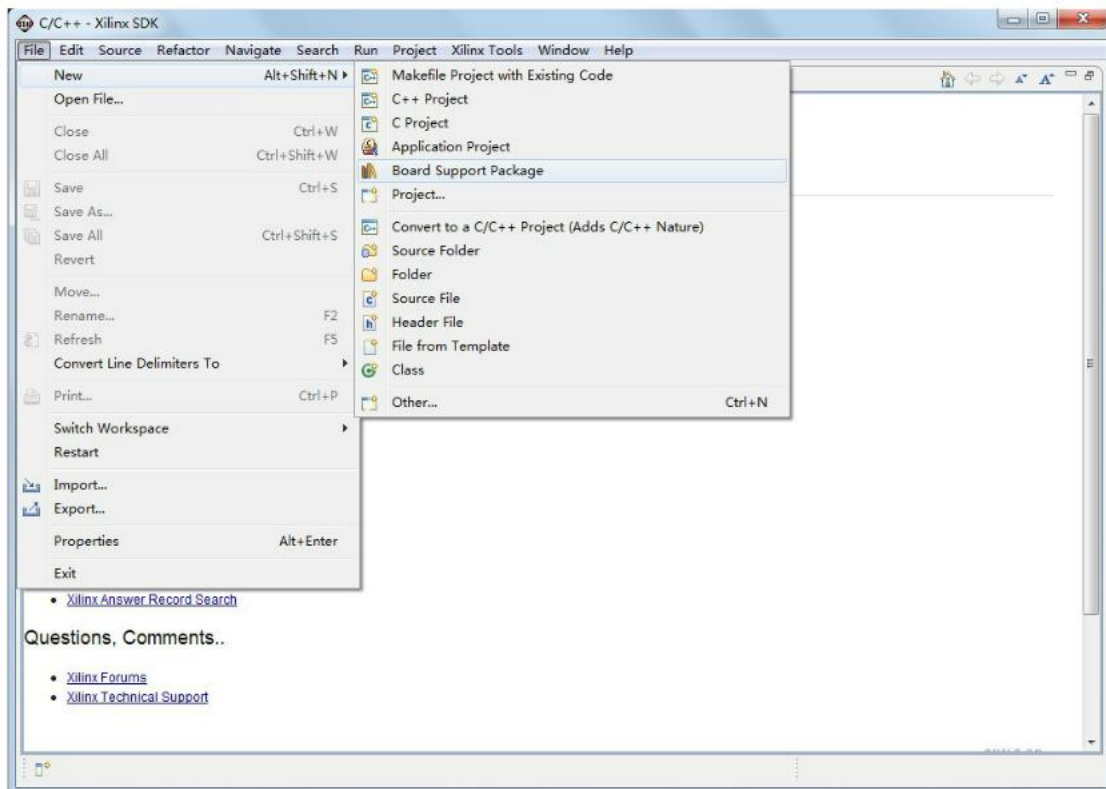
ISE 14.5 SDK 使用指南

现在网上找到的 EDK 工具的资料很多都是比较老旧的，现在贴出一个 ISE14.5 SDK 创建工程的例子。首先是将 XPS 生成的 bit 文件，bmm 文件和 xml 文件复制出来放在一个文件夹中，然后就可以打开 SDK 创建 SDK 工程了。

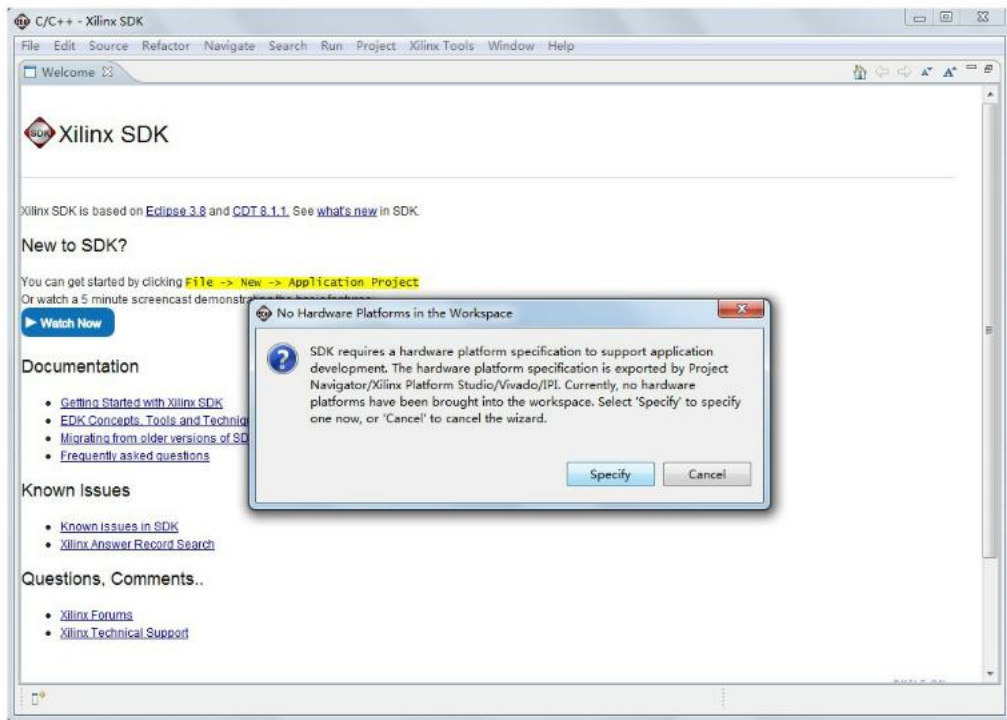
Step1 选择 workspace 也就是工程路径



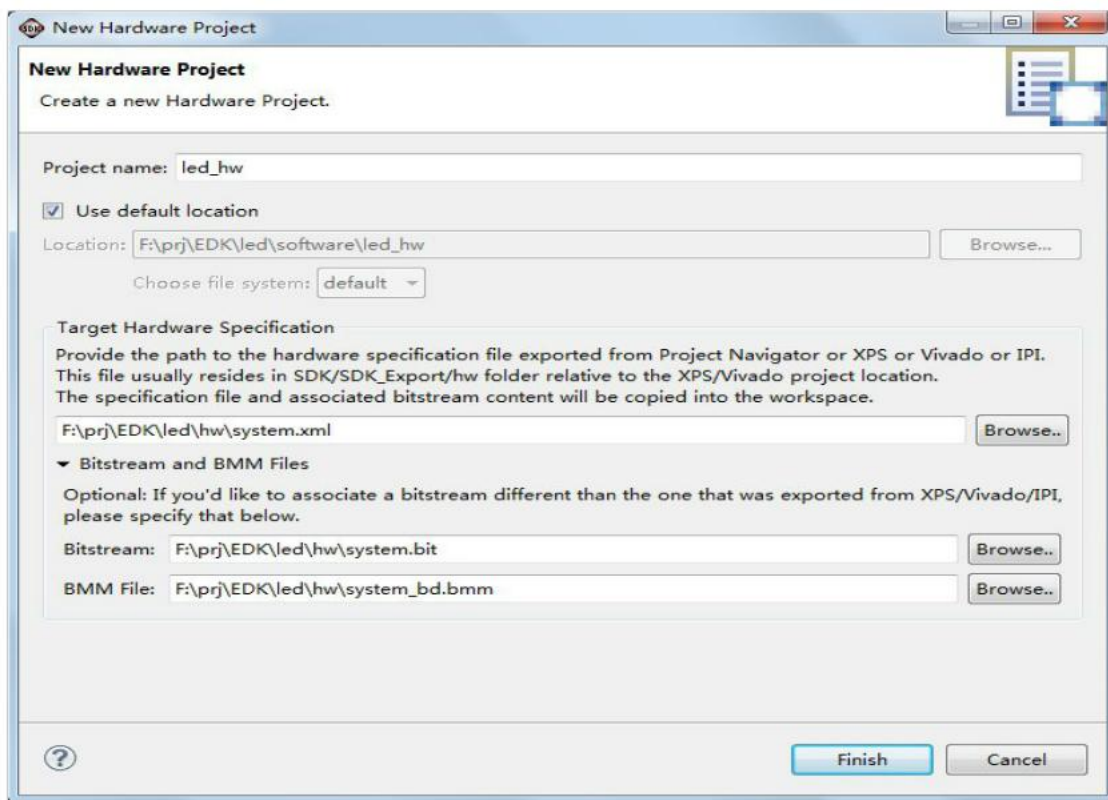
Step2 创建一个新的 Broad support package，后面的软件工程都是搭建在这个 BSP 基础上的。



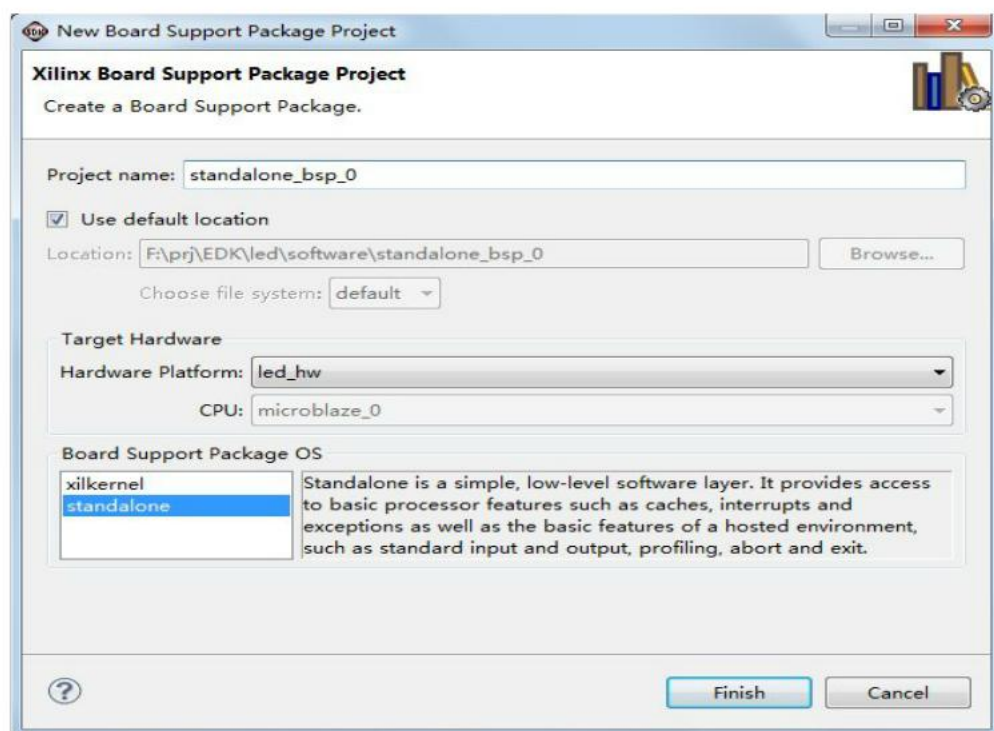
Step 3 要求指定硬件平台信息，选择 Specify



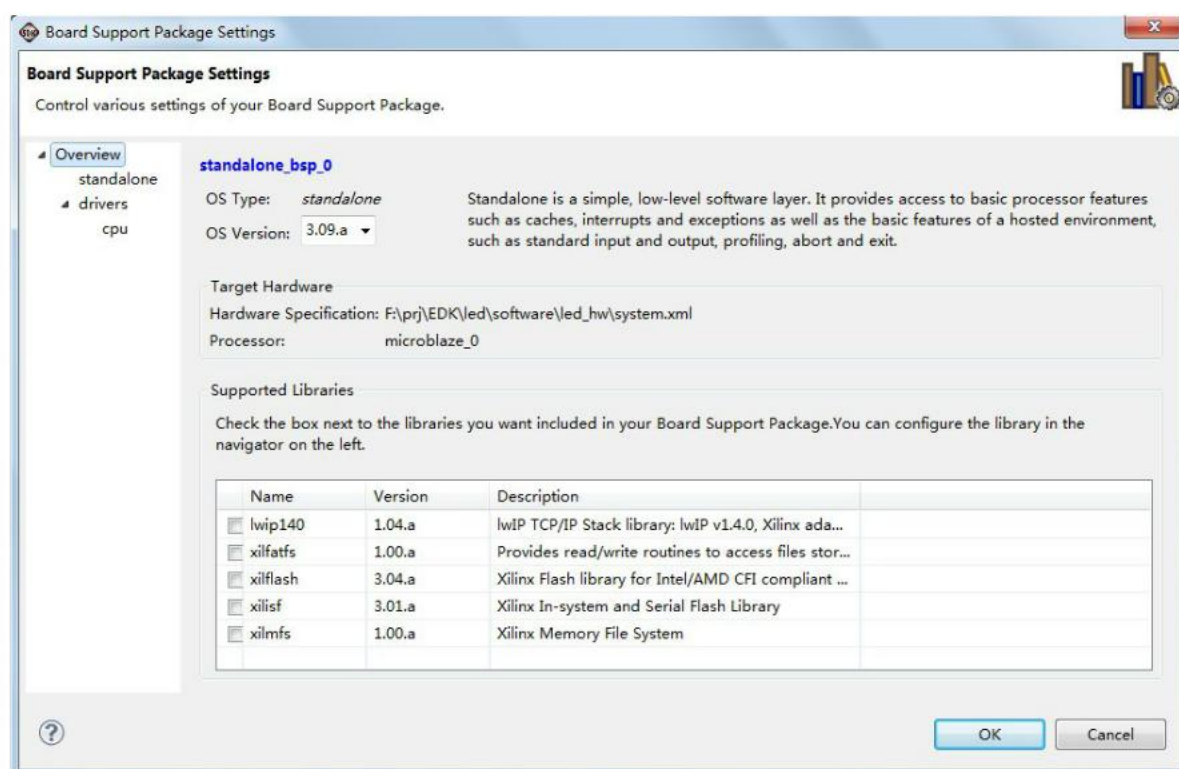
Step 4 给硬件平台工程起名字及将 xml bit 和 bmm 文件的存放路径指给软件



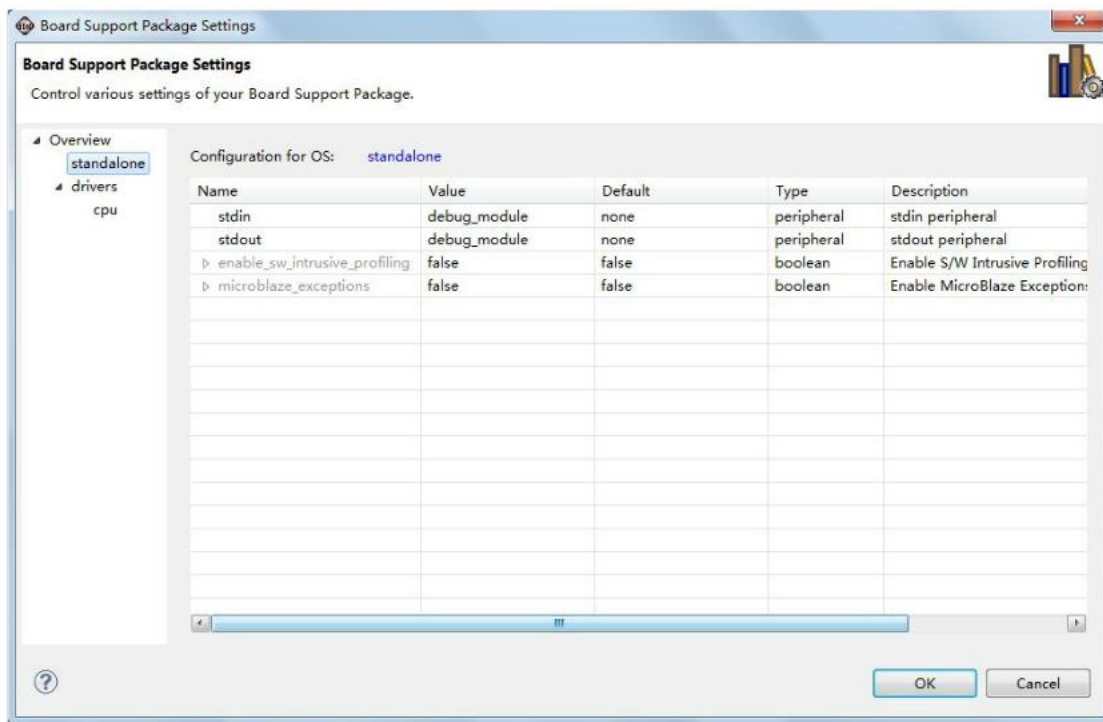
Step 5 这里选用默认的就可以了



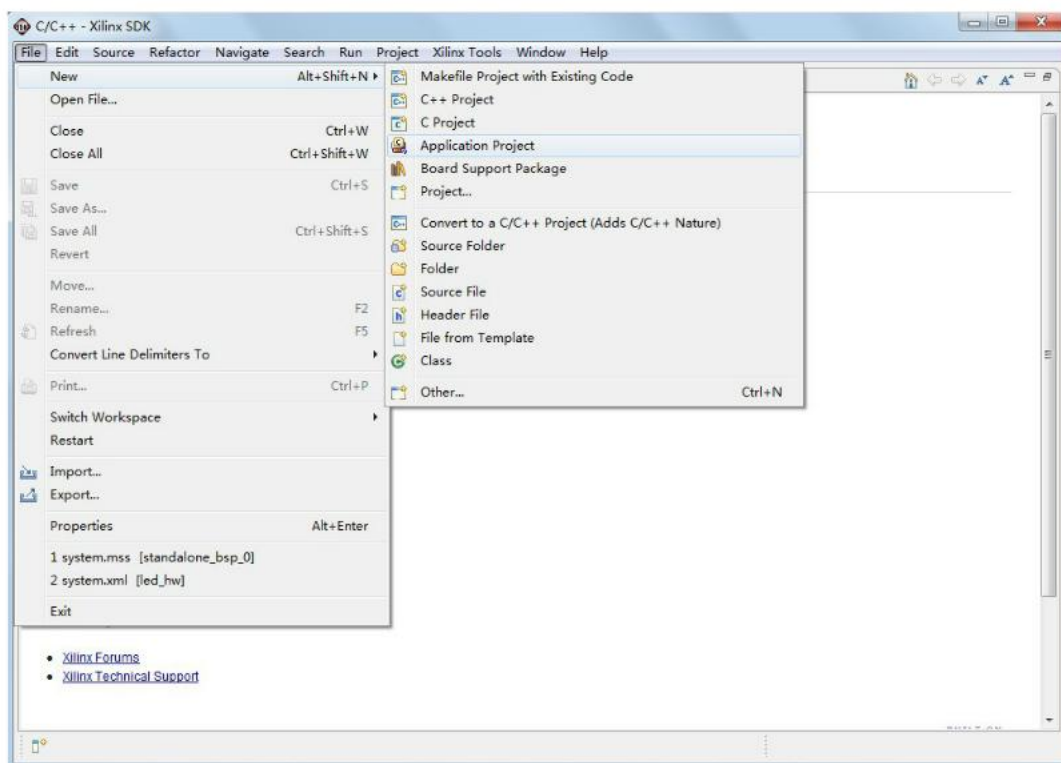
Step 6 根据需求配置 BSP 这里选择默认



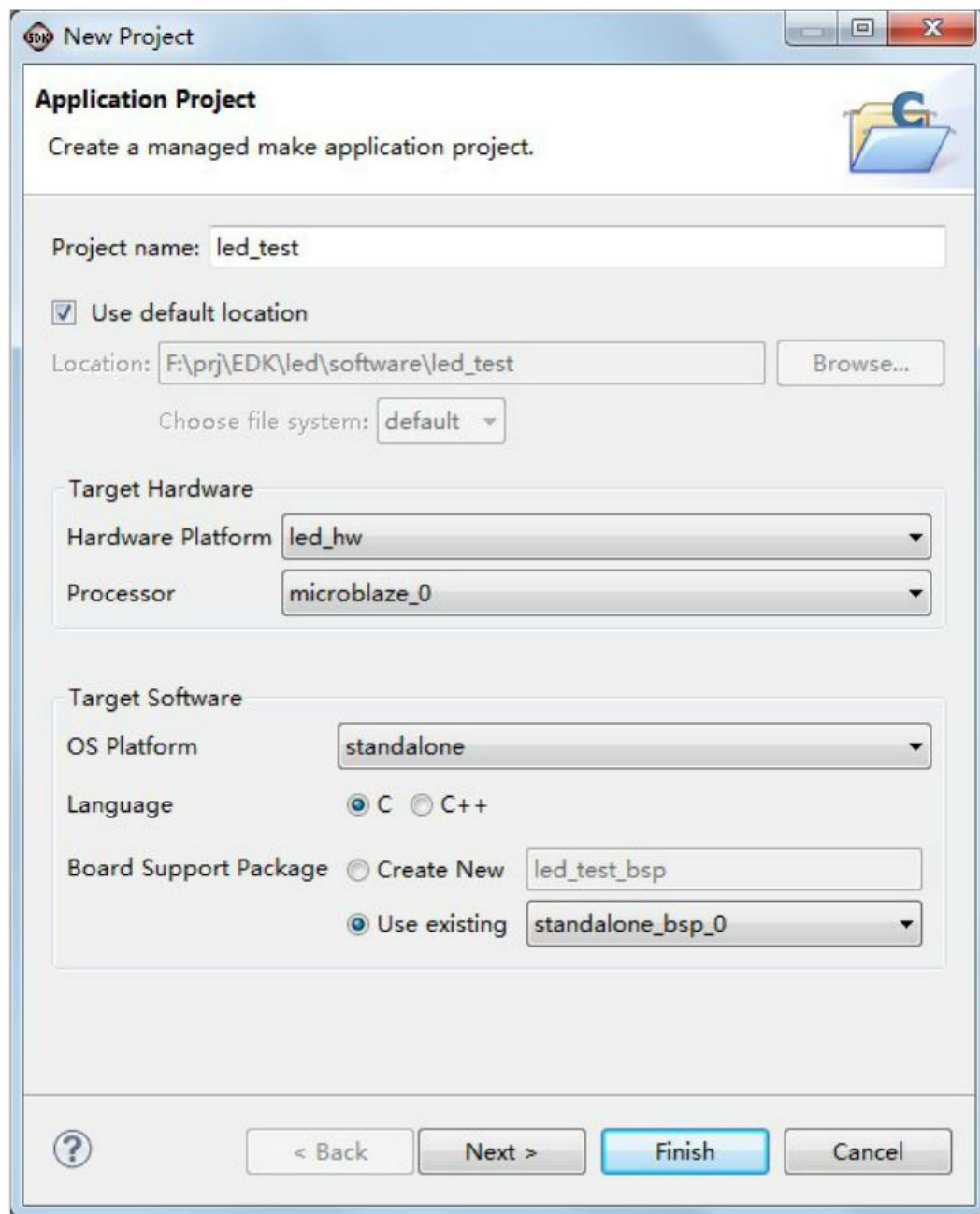
Step 7 BSP 中的配置，这里选择默认



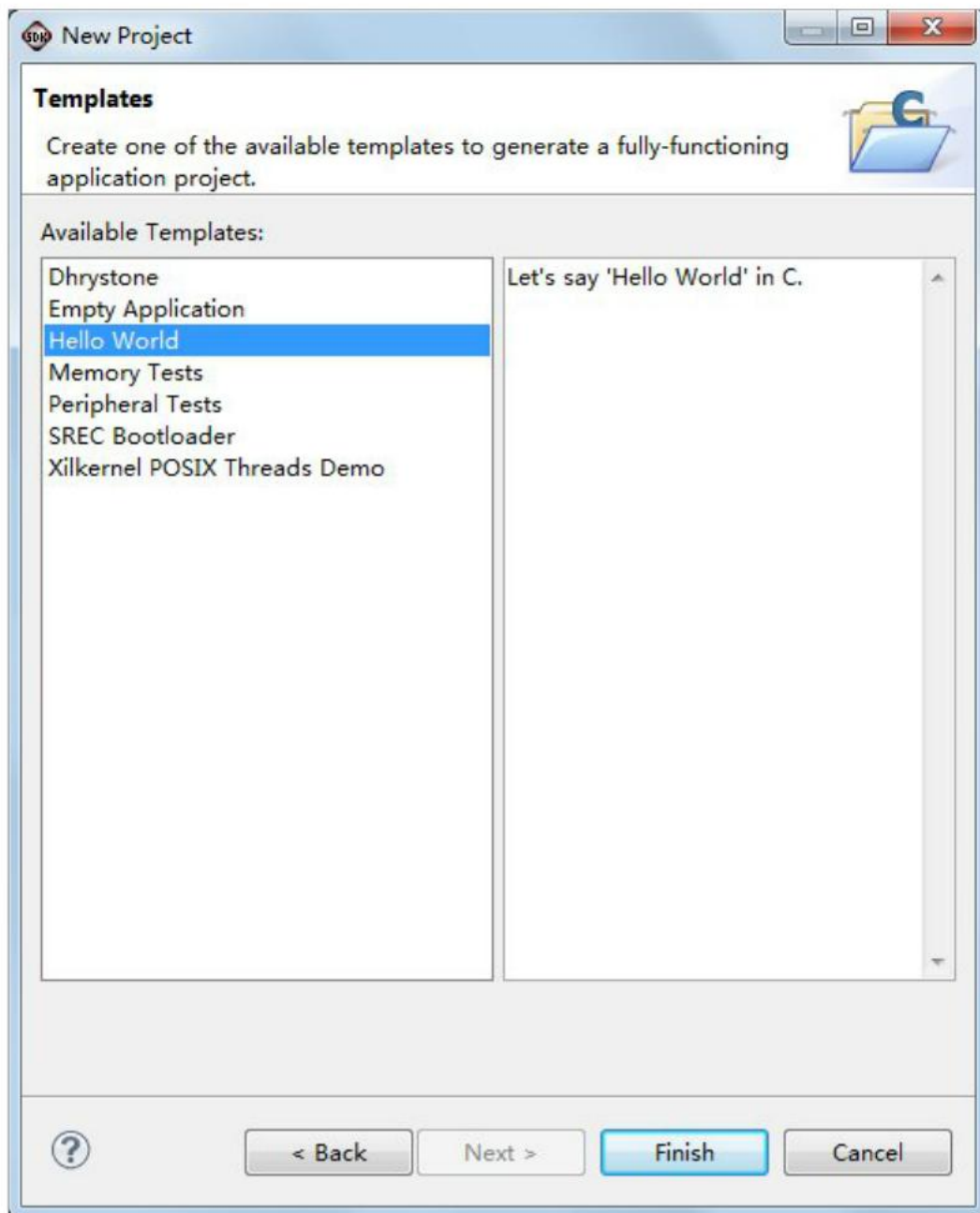
Step 8 创建完硬件平台及 BSP 之后，需要新建应用工程



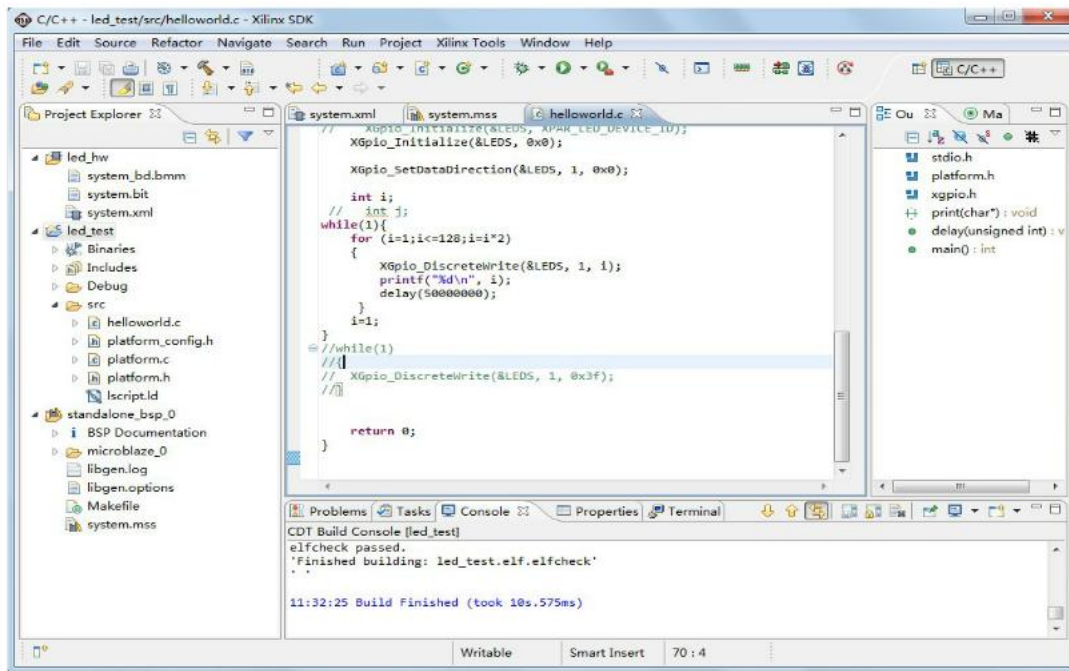
Step 9 给应用工程起个名字，及选择 exiting standalone BSP



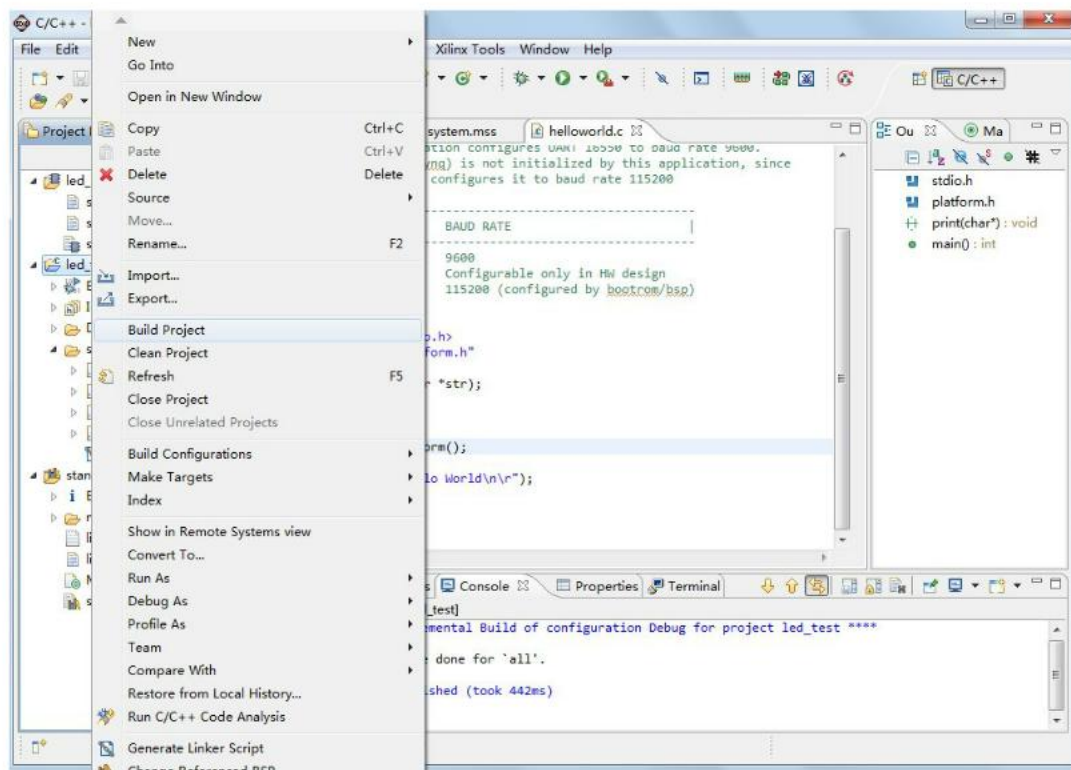
Step 10 选择一个最简单的 hello world 工程



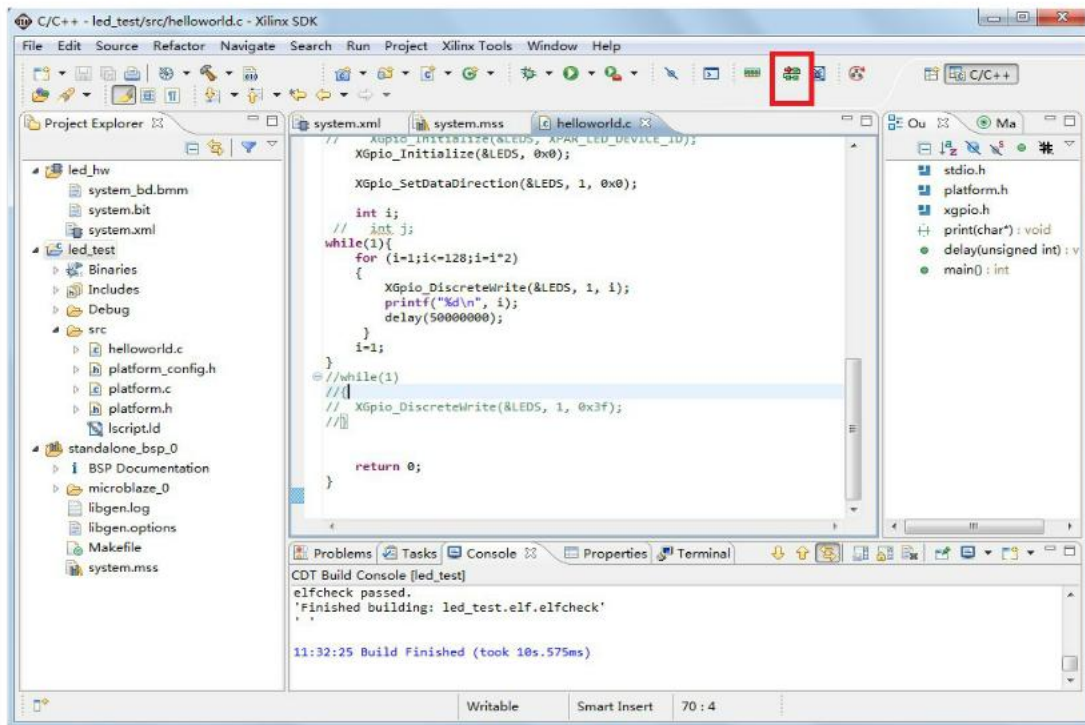
Step 11 在 Hello world 工程的基础上改成点亮 LED 的工程



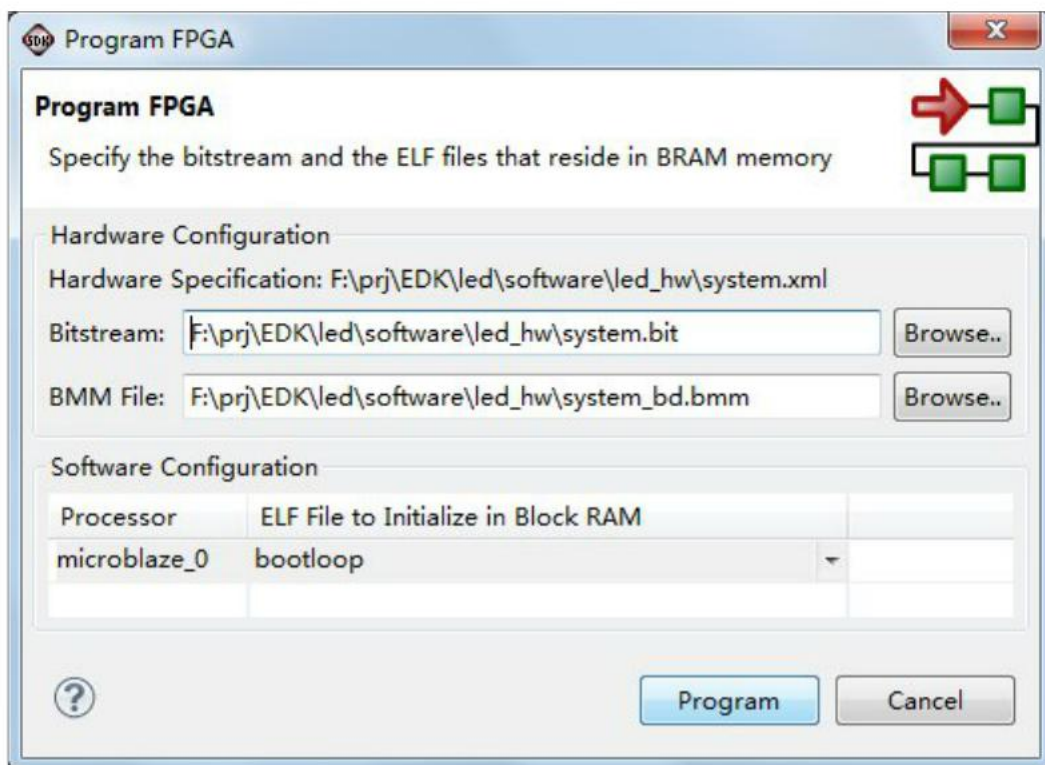
Step 12 右键点击 LED_TEST 应用工程，选择 Build Project 来编译工程。或者用 ctrl + b 来编译



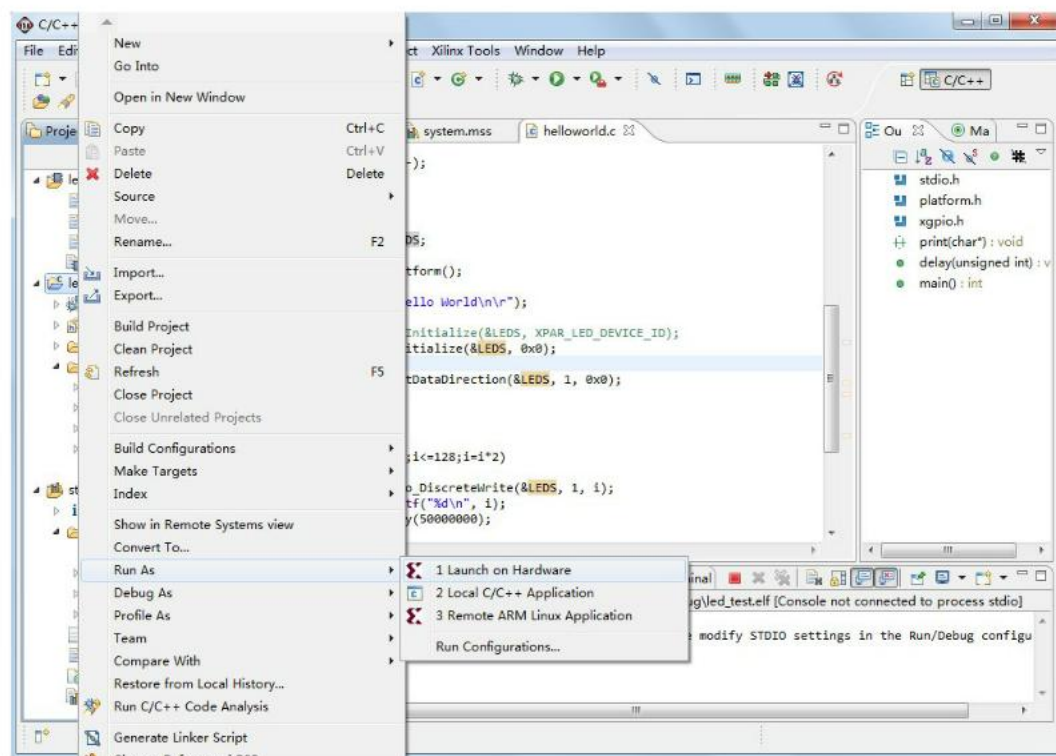
Step 13 点击红框中的按钮，将 bit 文件和 bmm 文件配置到 FPGA 中



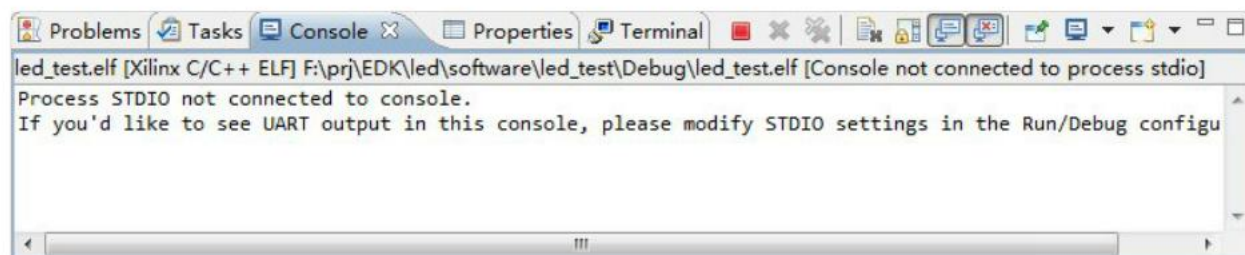
Step14 选择需要配置的 bit 文件和 bmm 文件



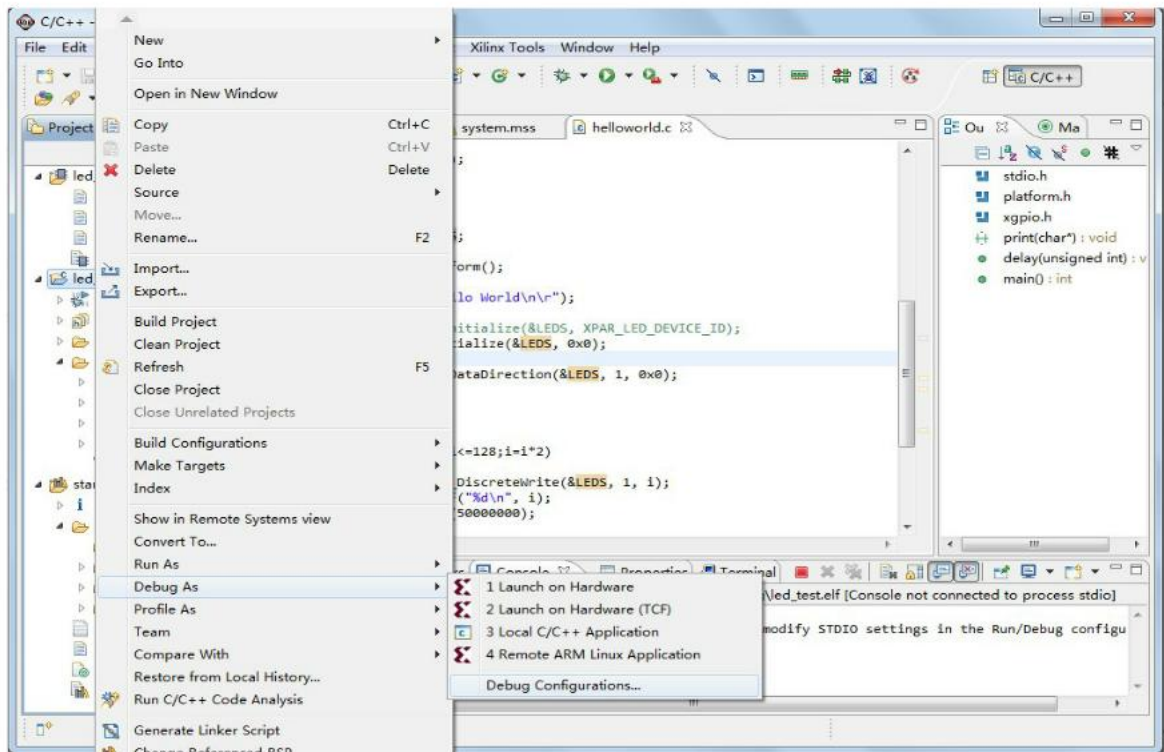
Step 15 右键点击 led_Test 应用工程，选择在 FPGA 上跑此工程。第一次必须在此点击，否则报错



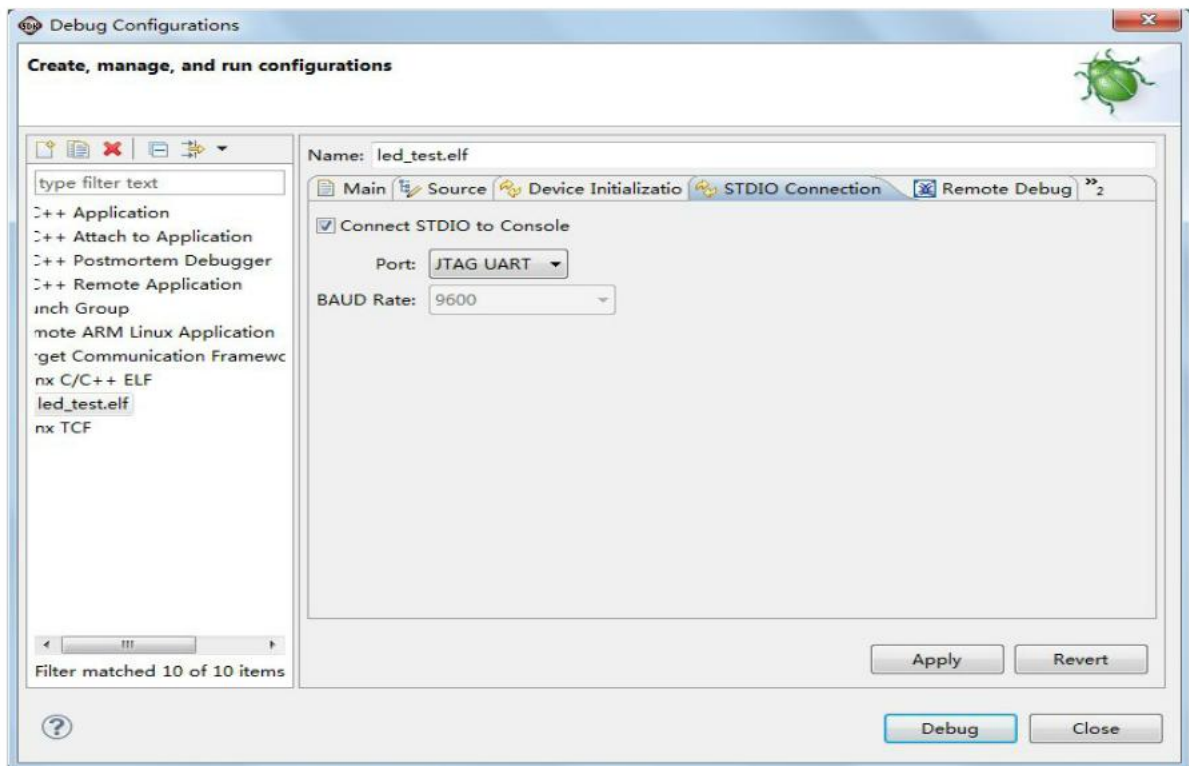
Step 16 提示 STDIO 调试串口没有打开



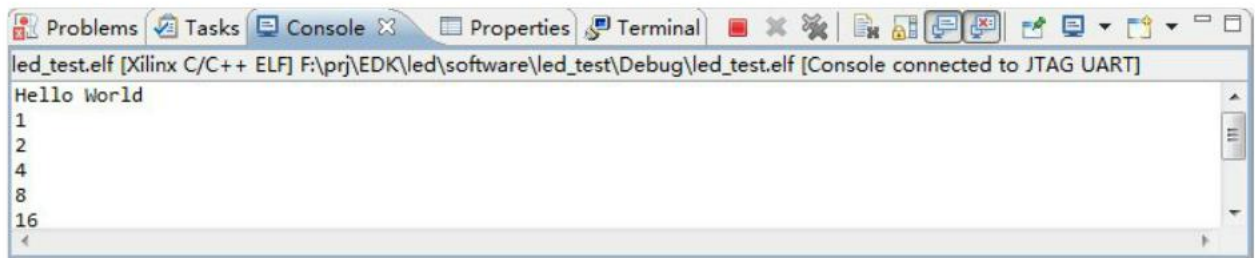
Step 17 打开 JTAG 自带的串口



Step 18 选择 STDIO Connection 勾选 Connect STDIO to Console。这样就打开了串口



Step 19 打开串口之后可以看到输出结果了。



The screenshot shows a console window with the following content:

```
led_test.elf [Xilinx C/C++ ELF] F:\proj\EDK\led\software\led_test\Debug\led_test.elf [Console connected to JTAG UART]
Hello World
1
2
4
8
16
```