FreeRTOS port for Xilinx Zynq devices

This port was created using version 14.4 of the Xilinx ISE Design Suite and was tested on a Zynq ZC702 board. It is based on version 7.0.2 of FreeRTOS. This port is tested with default Zynq ZC702 system with a CPU frequency of 667MHz in JTAG boot mode. This port utilizes SCUTIMER for generating tick interrupts and UART for displaying messages on console. The frequency of SCUTIMER is half of the CPU frequency.

Note - This FreeRTOS port is not supported by Xilinx Technical Support and verification of this port is limited. This port is tested with v3_08a version of standalone BSP and will get updated for later releases. If the user needs to use later versions of standalone BSP released in 14.5, he has to modify the version defined for the parameter standalone_version in the file <Xilinx_Zynq>/sw/repo/freertos_zynq_v1_01_a/data/freertos_zynq_v2_1_0.tcl and version against DEPENDS option in the file <Xilinx_Zynq>/sw/repo/freertos_zynq_v1_01_a/data/freertos_zynq_v2_1_0.mld.

Using FreeRTOS in the Xilinx SDK environment

A stand-alone board support package (BSP) is a library generated by the Xilinx SDK that is specific to a hardware design. It contains initialization code for bringing up the ARM CPUs in Zynq and also contains software drivers for all available Zynq peripherals. However it is not FreeRTOS aware.
The FreeRTOS port provided in this package extends the stand-alone BSP described above to also include FreeRTOS source files. After using this port in a Xilinx SDK environment, the user gets all the FreeRTOS source files in a FreeRTOS BSP library. This library uses the Xilinx SDK generated stand-alone BSP library. None of the standalone drivers included in the stand-alone BSP library is thread-safe. The demo applications provided as part of this package are based on the FreeRTOS BSP.
The FreeRTOS package can be downloaded from Xilinx tab under the community forums.

The Xilinx_Zynq directory in this package has the following structure –

```
Xilinx_Zynq
  ¦
  +-sw      Repository used to integrate FreeRTOS related files and related apps in to SDK
    ¦
    +- repo
          ¦
          +- bsp        FreeRTOS Source files
          ¦
          +- sw_apps Contains Apps for Hello World, Blink LED using Semaphore and Blink LED using Mutex
```

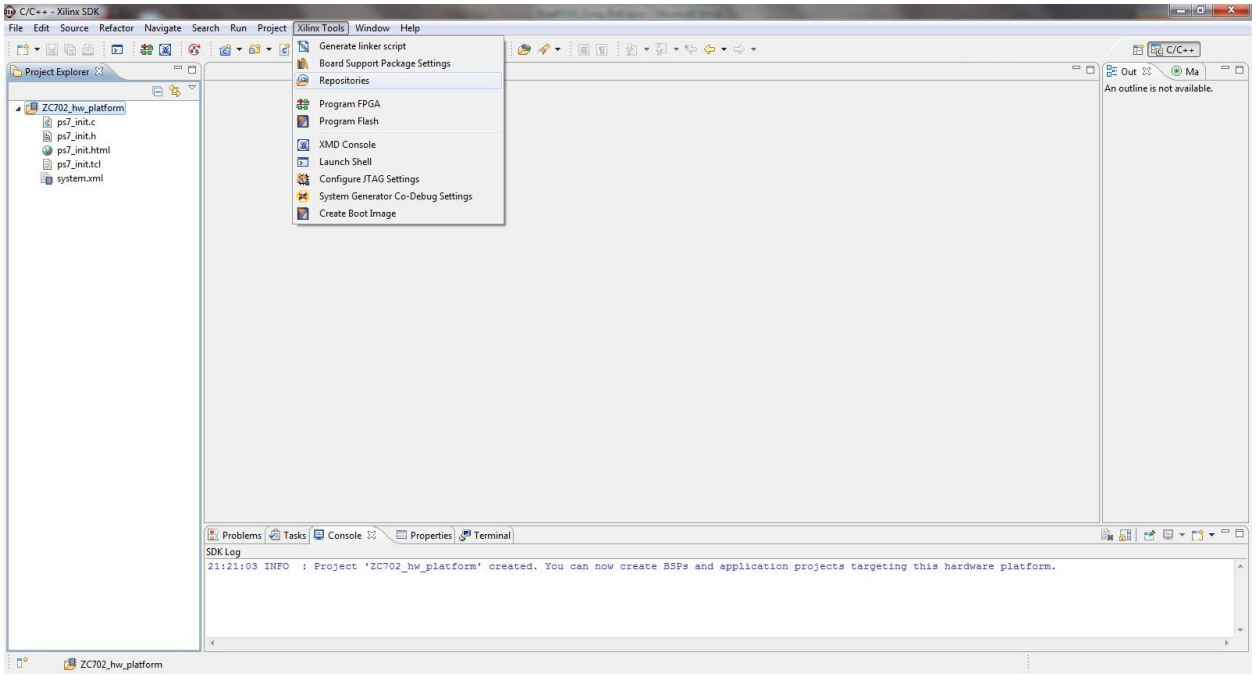
- The **bsp** folder contains all FreeRTOS source files. These source files include the generic FreeRTOS source (that this port has not changed) and Zynq related source files (which are newly written as per the Zynq hardware requirements).

- The **sw_apps** contains demo applications that the user can run to test the FreeRTOS port. It contains a simple hello world application that prints messages from multiple tasks. It contains two LED examples that blink a LED on ZC702 board, one using mutex and the other using timer and semaphore.

Create FreeRTOS Application and BSP using Xilinx SDK

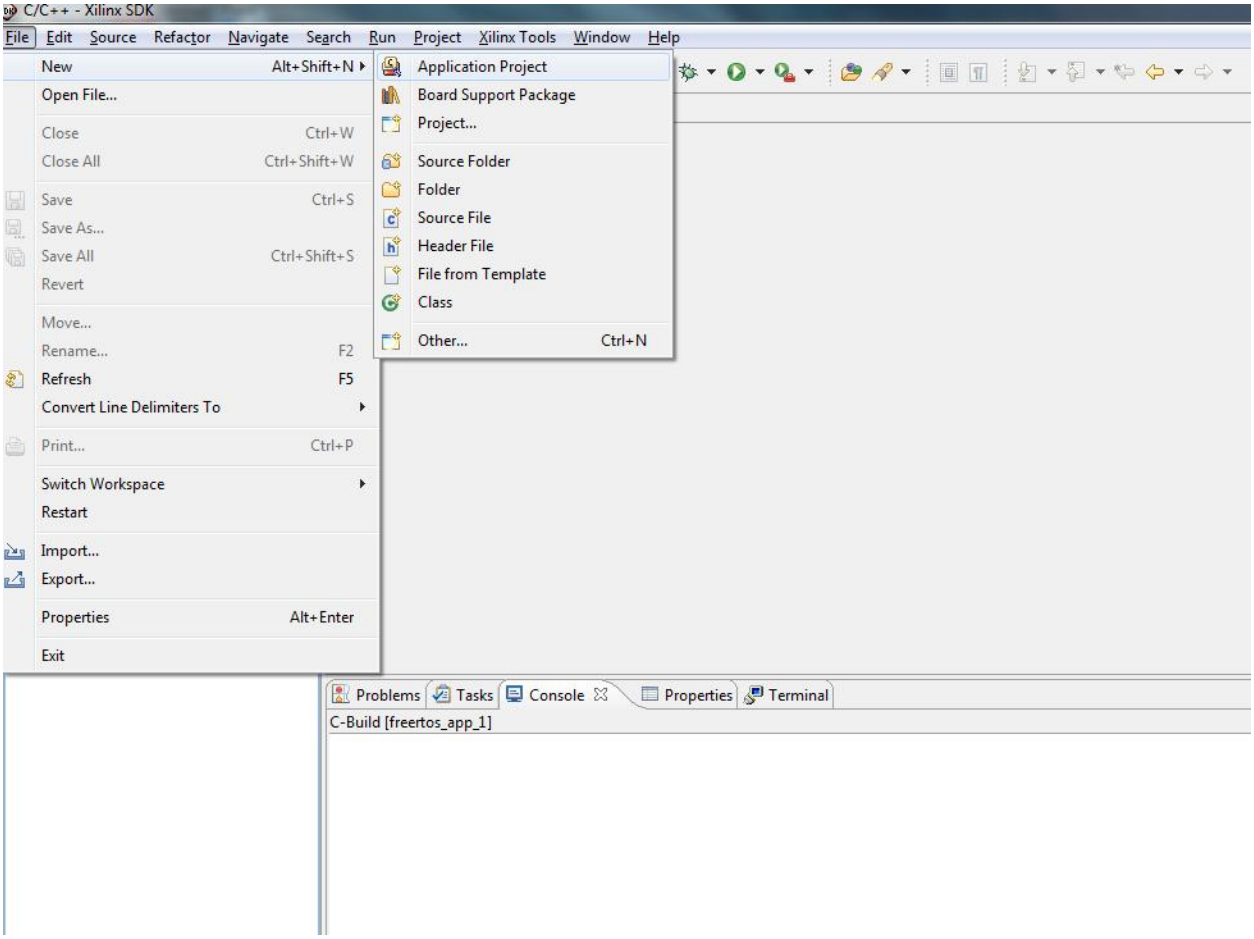
- Extract the zip file available in the package to get the Xilinx_Zynq directory.
- Create a folder for SDK workspace with a suitable name, say *sdk_projects*.
- Open SDK with *sdk_projects* as workspace.

Select Xilinx Tools -> Repositories. Preferences Window pops up.



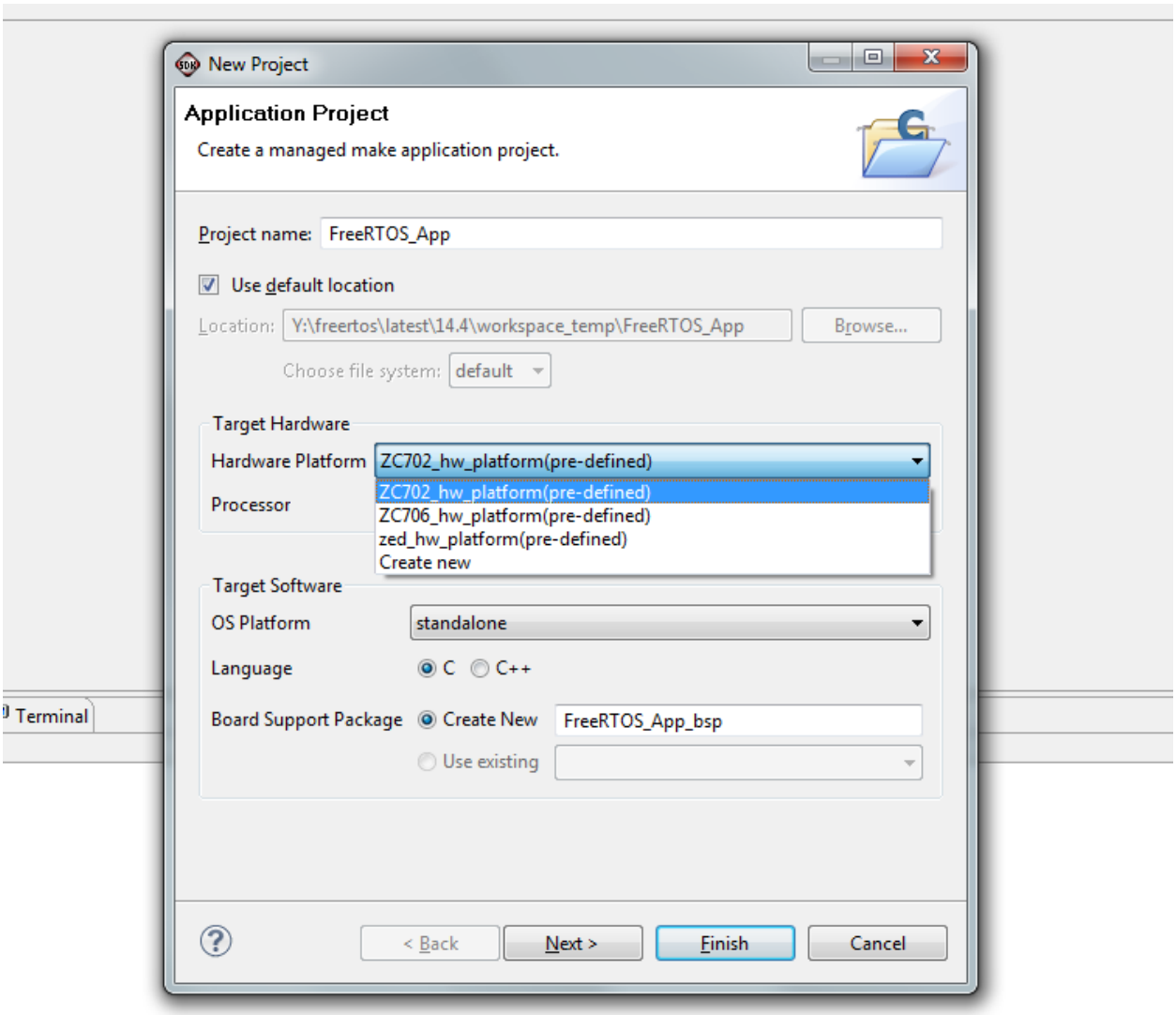
Click New under Local Repositories section and give the path to Xilinx_Zynq/sw/repo/ directory.
Click Rescan Repositories, then select Apply and then OK. This will ensure that the Xilinx SDK knows about the FreeRTOS BSP being available to it.

Choose File -> New -> Application Project -> Select.

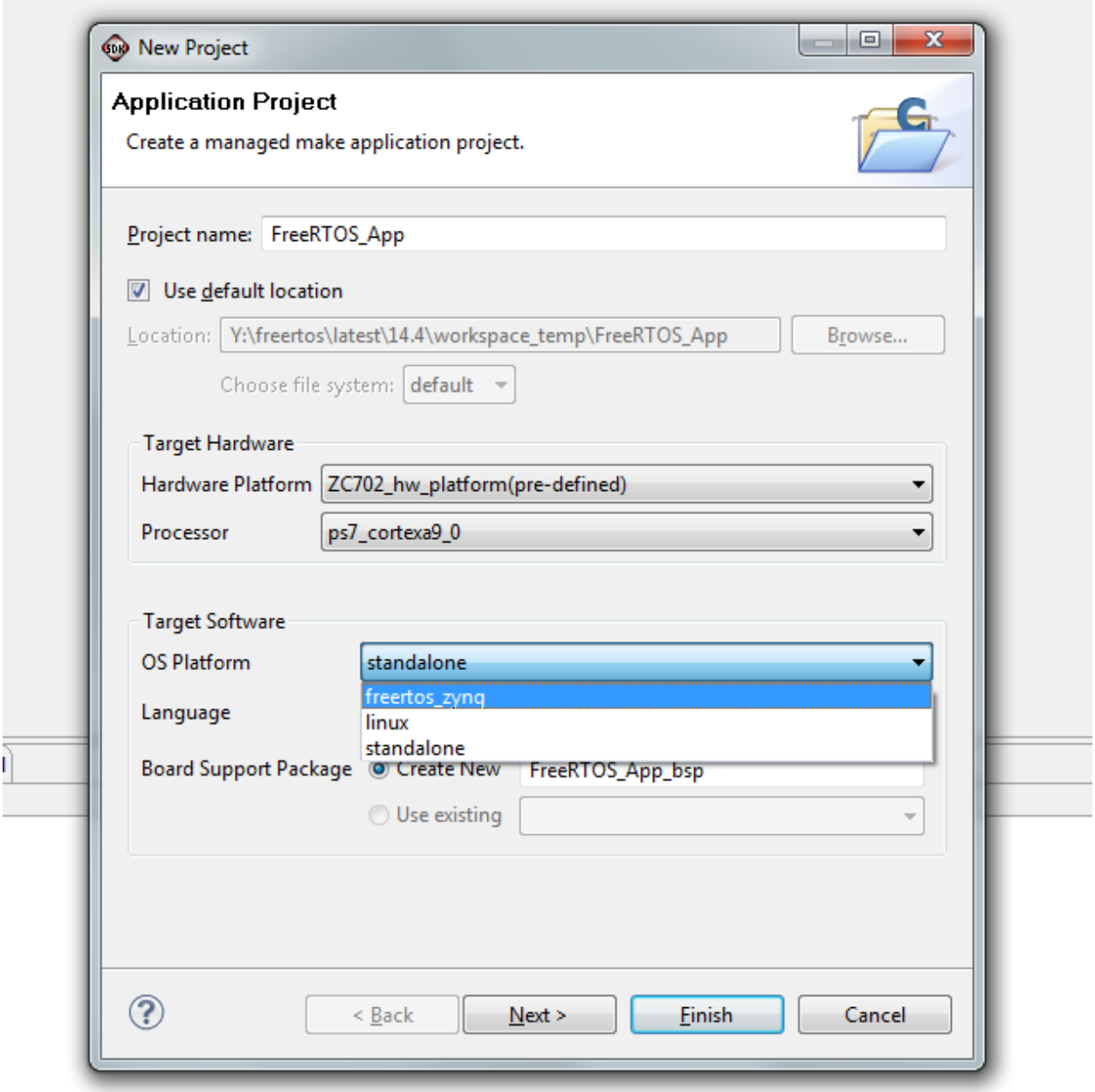


New Application Project window opens up. Provide Project Name.

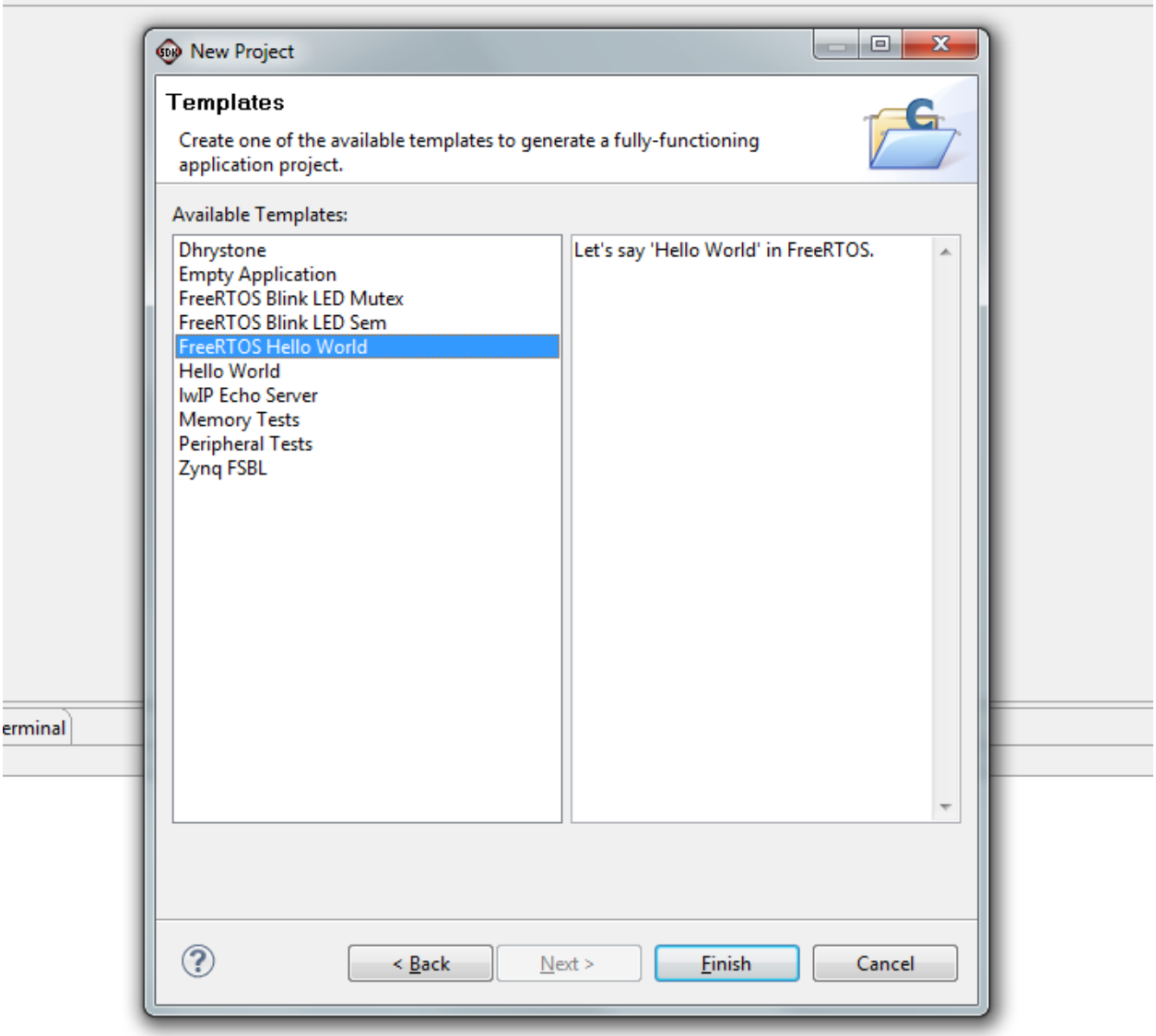
Under Target Hardware tab, choose a Hardware Platform from dropdown list against Hardware Platform attribute. Choice can be made to use pre-defined hardware platforms or create new hardware project, say ZC702_hw_platform. Choose the processor for which the application should be targeted.



Under Target Software tab, select freertos_zynq as OS Platform. Name for Board Support Package will be populated based on the application project name. Click Next.



Select any of the available FreeRTOS applications, say FreeRTOS Hello World. Click Finish. Three sdk projects – FreeRTOS Hello World application with the provided project name, board support package and ZC702 hardware platform project will be created.



Create other FreeRTOS APPS

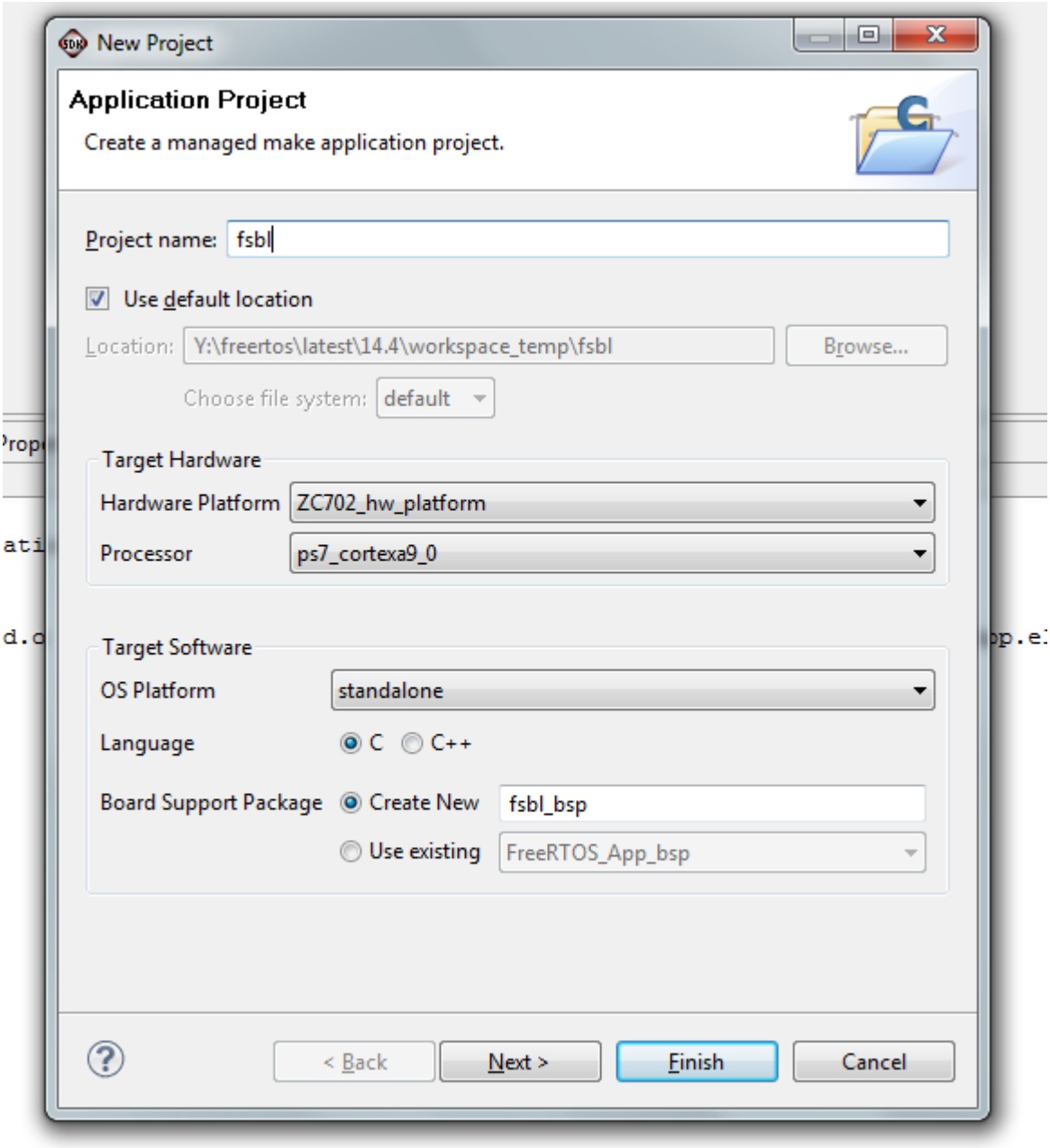
- Choose File -> New -> Application Project -> Select.
- Give Project Name, choose OS Platform as FreeRTOS Zynq and for Board Support Package select the radio button ‘Use existing’, which points to already created BSP project. Click Next.
- Choose one of the FreeRTOS Applications available. Click Finish.

FreeRTOS Hello World – It simply creates two tasks with a print statement in each and of equal priorities. The prints should be observed on terminal according to scheduling policy. Expected output is prints from both tasks one after another.

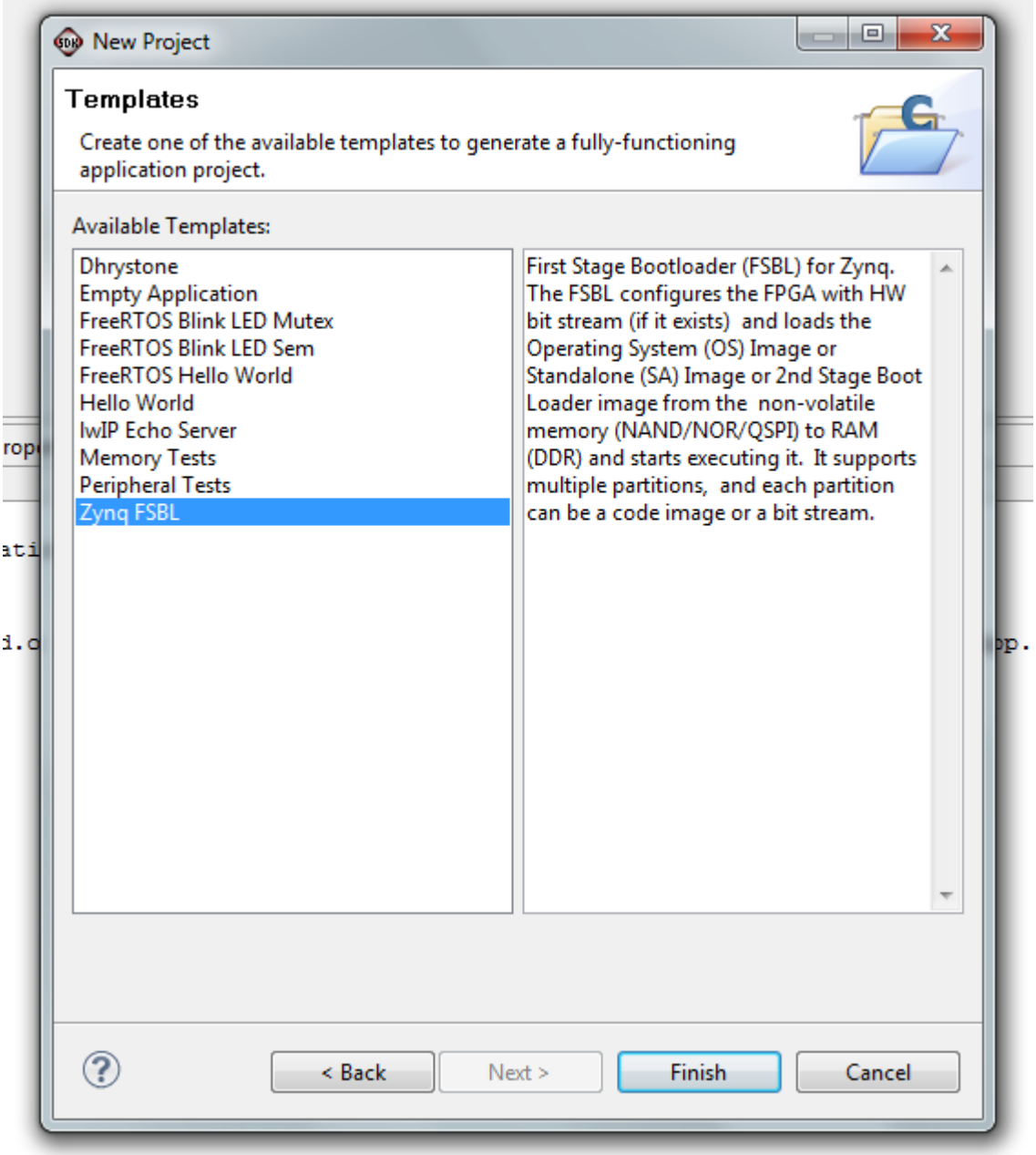
FreeRTOS Blink LED Sem – It creates a task which toggles LED while waiting on a semaphore. A timer signals the semaphore when expired and duration to expire is based on TIMER_PERIOD macro. Expected output is LED DS23 toggles ON & OFF.
FreeRTOS Blink LED Mutex – It creates two tasks, one sets the LED state to high and the other to low. Each task calls a delay using BLINK_PERIOD value which defines the duration to retain a particular state. The control is alternated between these two tasks using mutex.

Create First Stage Bootloader application (FSBL) for Zynq

Before running any application on the Zynq board, a FSBL application must be run to initialize the MIO pins and relevant clocks. FSBL is only needed if the user follows XMD flow and executing the application through SDK does not require FSBL.
Select File -> New -> Application Project -> Select. Provide Project Name as FSBL.
Under Target Hardware tab, choose a Hardware Platform from dropdown list against Hardware Platform attribute. Choice can be made to use pre-defined hardware platforms or create new hardware project. Choose the processor for which the application should be targeted.
Under Target Software tab, select standalone as OS Platform. Name for Board Support Package will be populated based on the application project name. Click Next.



Under Available Project Templates, choose Zynq FSBL. Click Finish.



Test Execution Procedure from XMD

Open an XMD shell [Make sure the Xilinx Platform Cable and UART cable and power to the board are properly setup]. Also open a terminal with 115200 8n1 settings for UART port.

- Switch on the board.
- Type “connect arm hw” and hit enter.
- Type “dow <path to Zynq fsbl elf file>” and hit enter.
- Type “con” and hit enter.
- Type “stop” and hit enter. [No prints are expected on the terminal during FSBL execution]
- Type “dow <path to freertos app elf file>” and hit enter.
- Type “con” and hit enter.

For the “FreeRTOS Blink LED Mutex “application, the user can see the DS23 LED on ZC702 board blinking with alternate print messages on console from two tasks with BLINK_PERIOD indicating the duration to retain the state.
For the “FreeRTOS Blink LED Sem” application, the user can see the DS23 LED on ZC702 board blinking with duration of LED state being configured using TIMER_PERIOD value.
For the “FreeRTOS Hello World”, the user can see prints on the console.

Known Issue –

The scutimer and scuwdt drivers will fail to link if any of the soft IP’s are used in PL part of Zynq. In order to fix it, one should add #include “xil_assert.h” in
<Install_Dir>/14.4/SDK/SDK/sw/XilinxProcessorIPLib/drivers/scutimer_v1_01_a/src/xscutimer_hw.h
<Install_Dir>/14.4/SDK/SDK/sw/XilinxProcessorIPLib/drivers/scuwdt_v1_01_a/src/xscuwdt_hw.h

References –

1. Zynq 7000 Extensible Processing Platform Technical Reference Manual – http://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf
2. Zynq 7000 Extensible Processing Platform Software Development Guide - http://www.xilinx.com/support/documentation/user_guides/ug821-zynq-7000-swdev.pdf
3. Other docs – http://www.xilinx.com/support/documentation/zynq-7000_user_guides.htm