# FreeRTOS port for Xilinx Zynq devices

This port was created using version 14.1 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_05a version of standalone BSP and will get updated for later releases. If the user needs to use later versions of standalone BSP released in 14.2 or later, he has to modify the version defined for the parameter standalone_version in the file <Xilinx_Zynq>/sw/repo/freertos_zynq_v1_00_a/data/freertos_zynq_v2_1_0.tcl.

## 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 are 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 a simple Hello World example and a Blink LED example
```

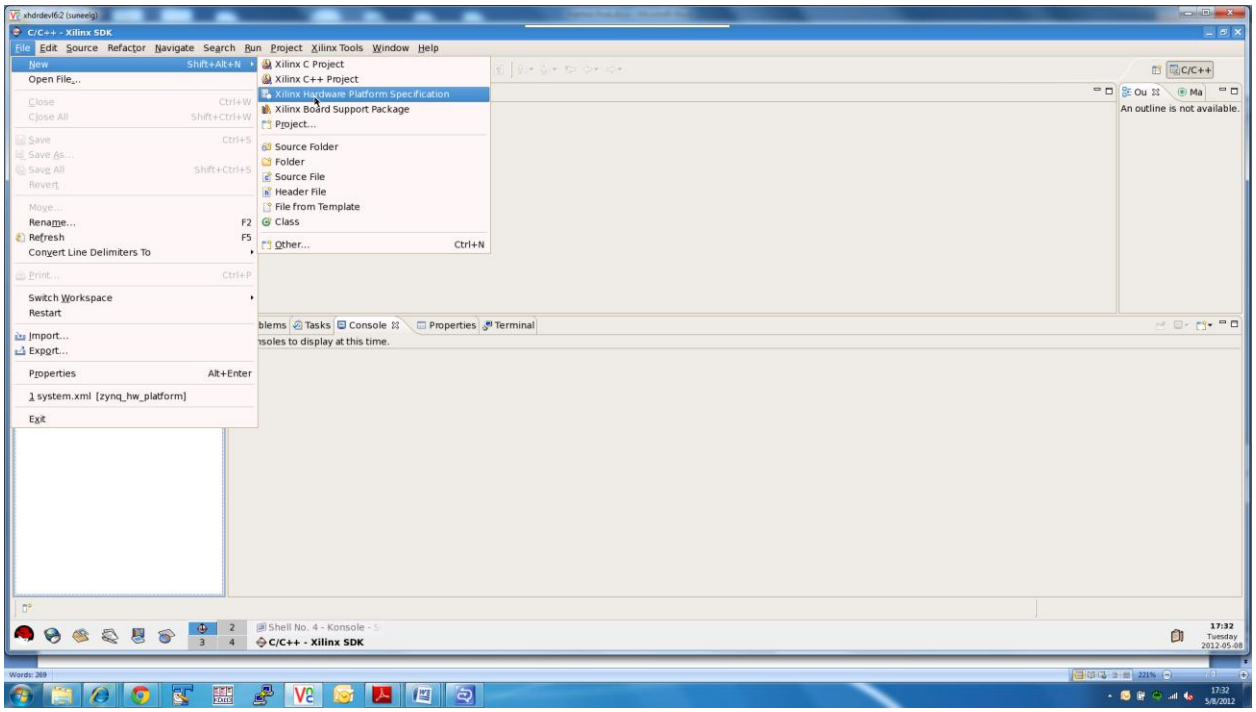
- The **bsp** folder contains all FreeRTOS source files. These source files include the generic FreeRTOS source (that this port has not changed) and Zynq related source files (which are newly written as per the Zynq hardware requirements).

- The **sw_apps** contains demo applications that the user can run to test the FreeRTOS port. It contains a LED example that blinks a LED on ZC702 board from multiple tasks and uses semaphores. It also contains a simple hello world application that prints messages from multiple tasks.

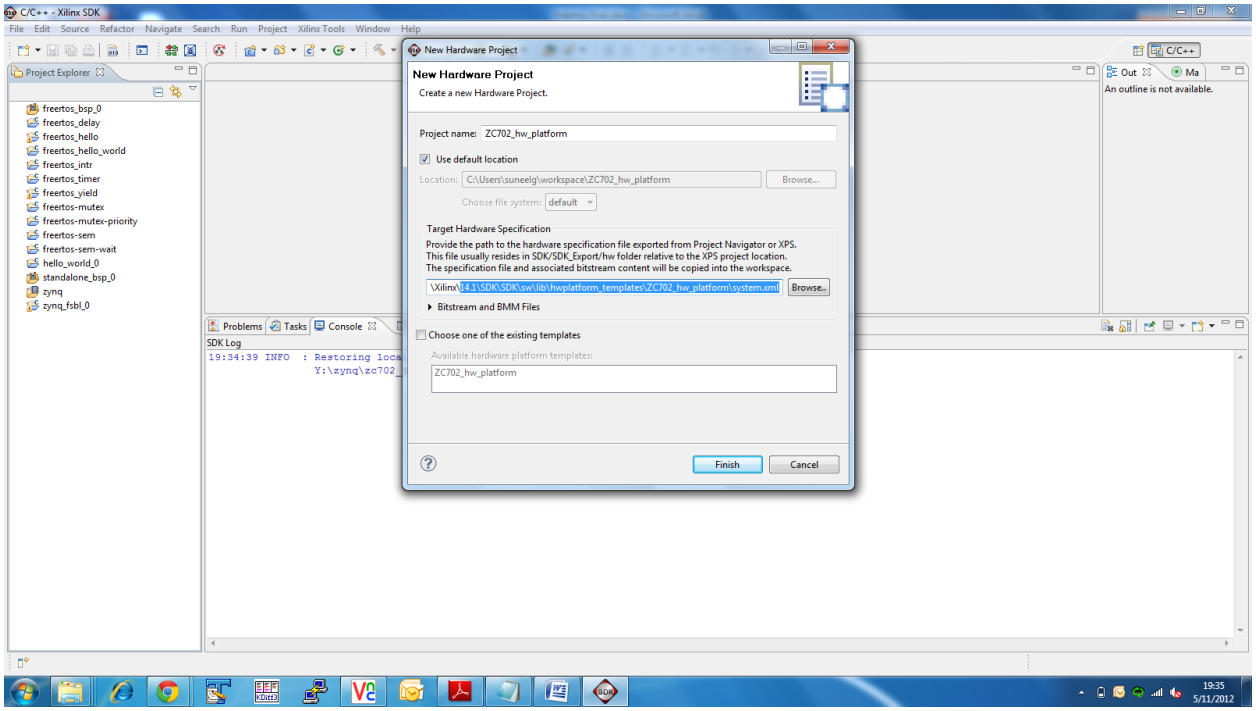
## Create FreeRTOS BSP using Xilinx SDK

- 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.

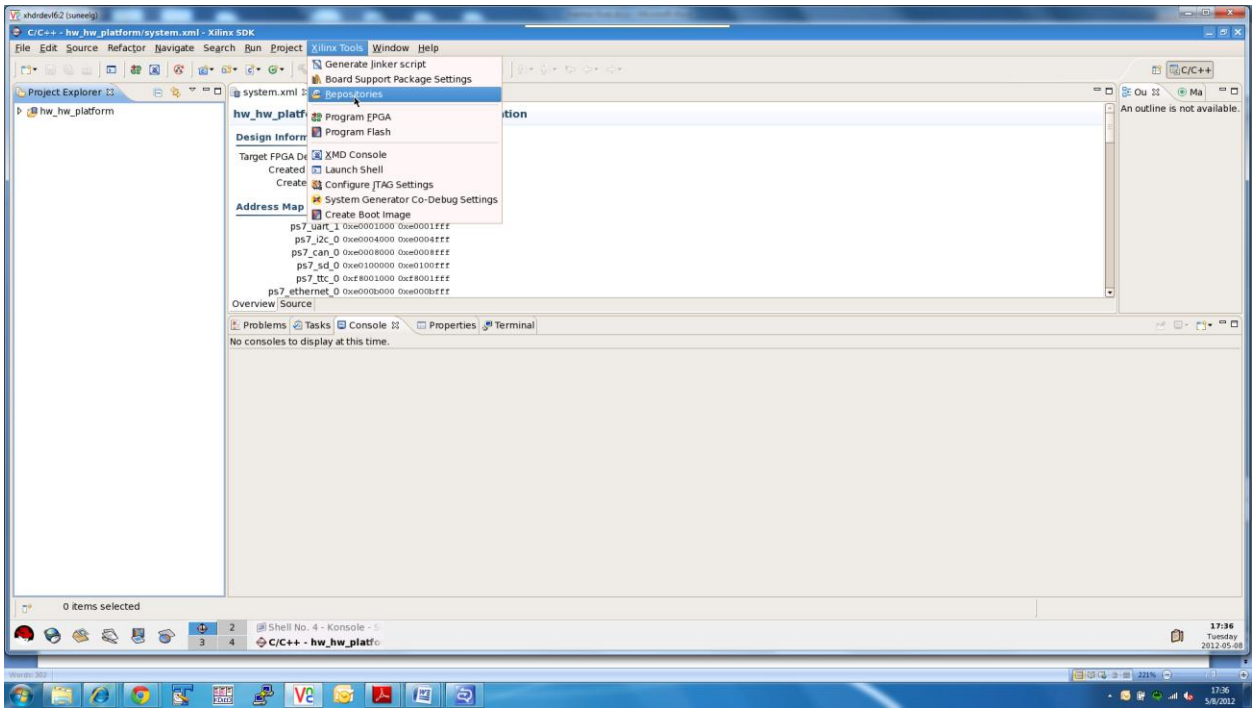
Select File -> New -> Xilinx Hardware Platform Specification. New Hardware Project window pops up.




Select an appropriate name for the hardware project and click browse to go to the <install_dir>\14.1\SDK\SDK\sw\lib\hwplatform_templates\ZC702_hw_platform in install directory of Xilinx Tools. Select system.xml to be used as the hardware description file.
Click Finish. This will create a hardware project in SDK with the given name and containing the files, system.xml, ps7_init.c, ps7_init.tcl, ps7_init.html and ps7_init.h.



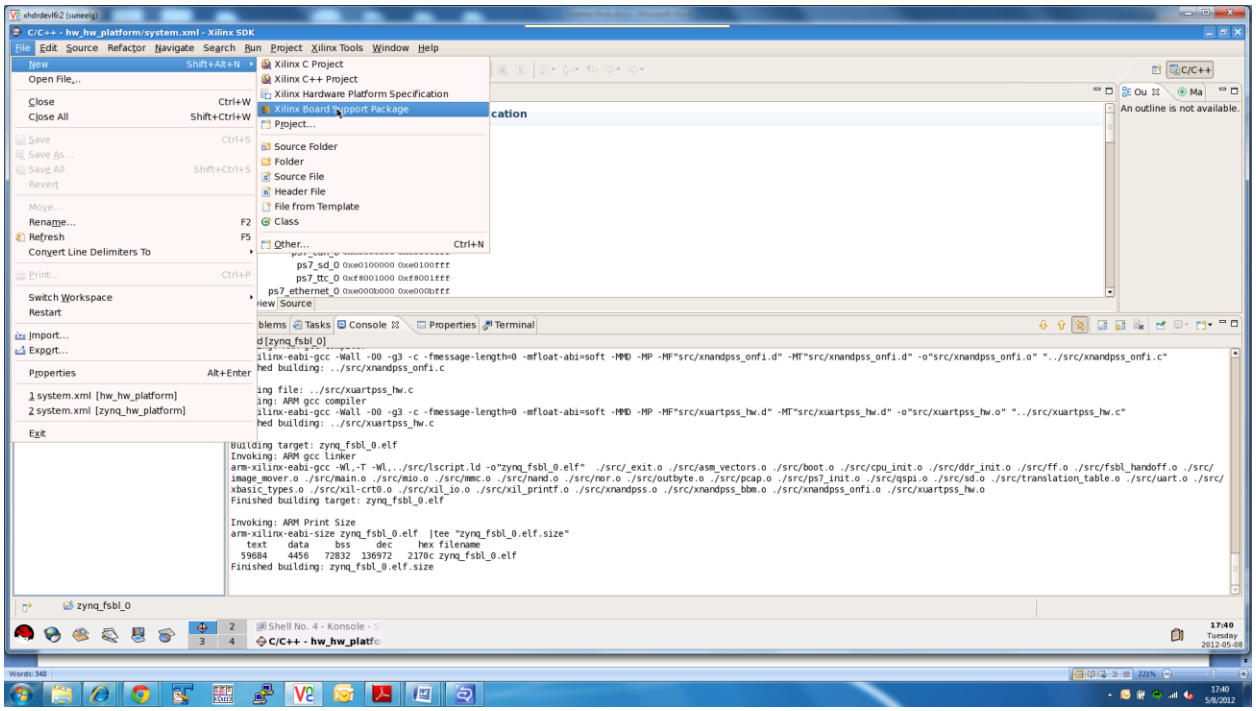

Select Xilinx Tools -> Repositories. Preferences Window pops up.



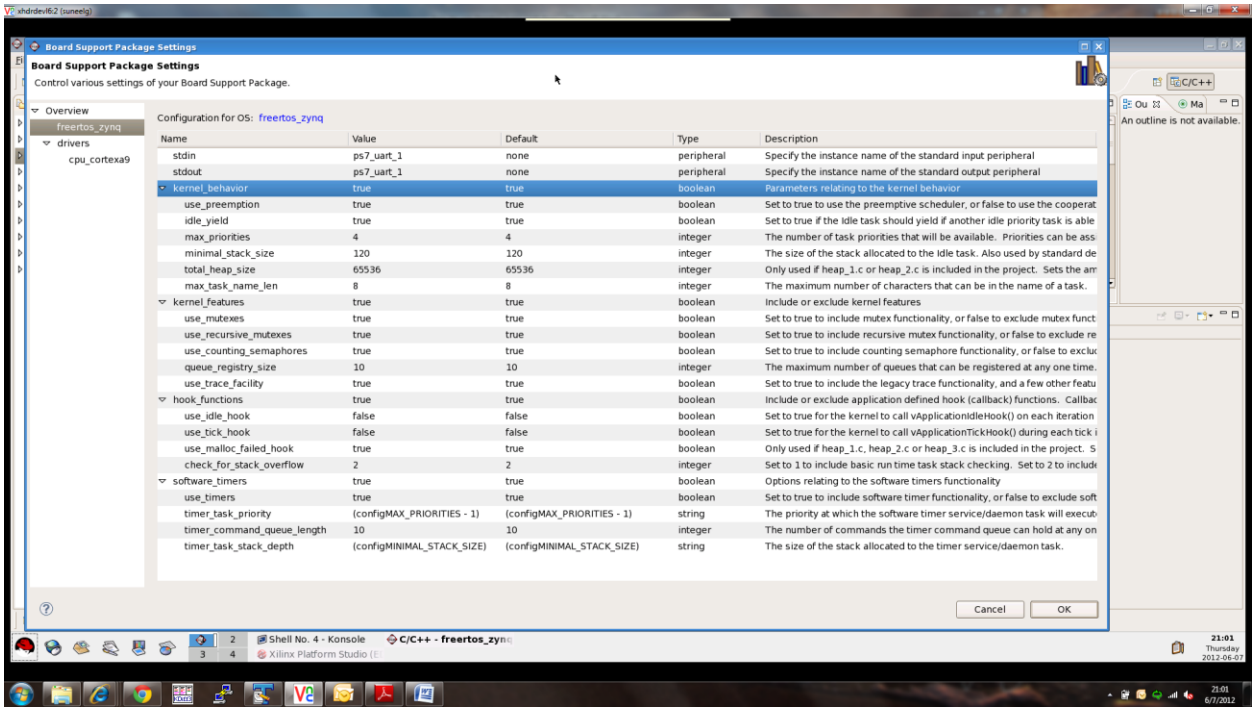

Click New under Local Repositories section and give the path to Xilinx_Zynq/sw/repo/ directory.

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.

Now select File->New-> Xilinx Board Support Package. New Board Support Package window pops up.




Select an appropriate name for the BSP in the top empty box. Down in this same window under the section “Board Support Package OS” select “freertos_zynq”. Then click “Finish”. The Board Support Package Settings window pops up where you can configure the features of FreeRTOS. Click “OK”.



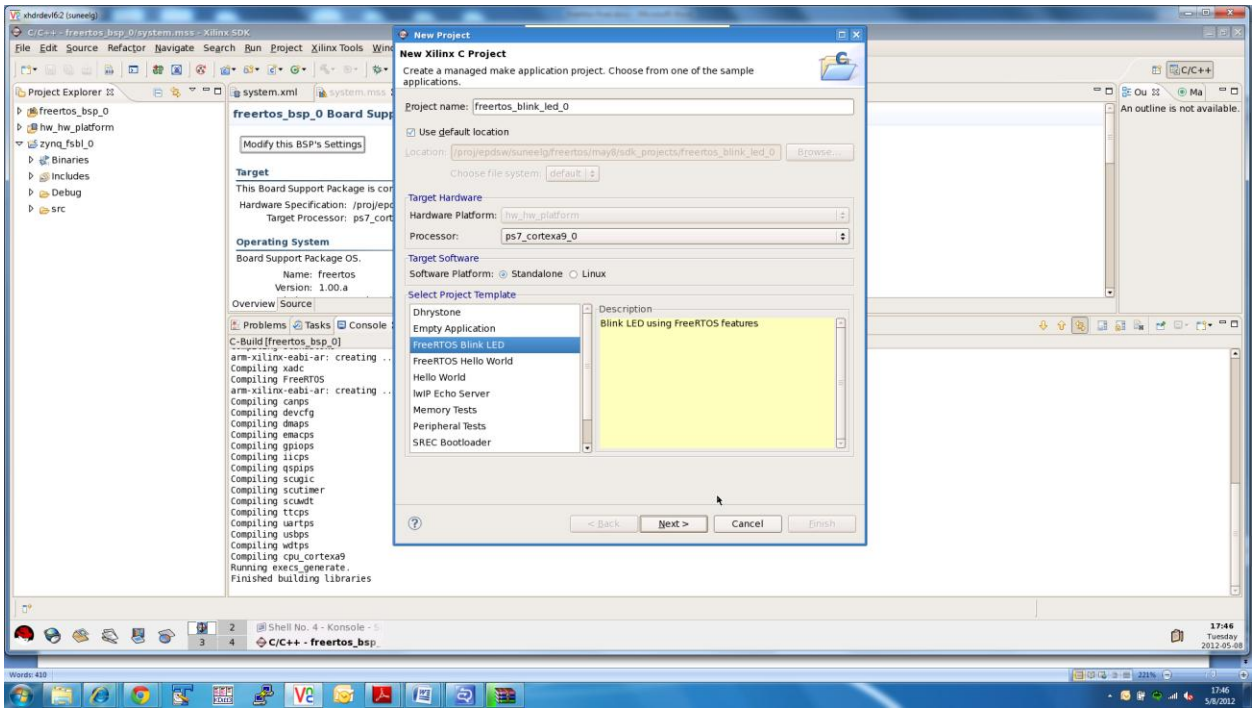

This will create and build a FreeRTOS BSP with the supplied name. This BSP directory will contain the FreeRTOS BSP (FreeRTOS related files), standalone BSP (v3_05_a) and software drivers for available peripherals in the hardware design for the Zynq ZC702 board.

## Create FreeRTOS APPS

- Choose File -> New -> Xilinx C Project -> Select.
- Under Select Project Template, choose FreeRTOS Hello World and click Next.
- Now Select the radio button corresponding to Target an existing BSP and select freertos_zynq.
- Click Finish. Repeat the same above 2 steps but choose FreeRTOS Blink LED to create second app.

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 – It simply creates two tasks with one task toggling LED to ON state and the other to OFF state. LED DS23 [MIO 10] on ZC702 board will turn ON/OFF. Expected output is LED toggles ON & OFF.

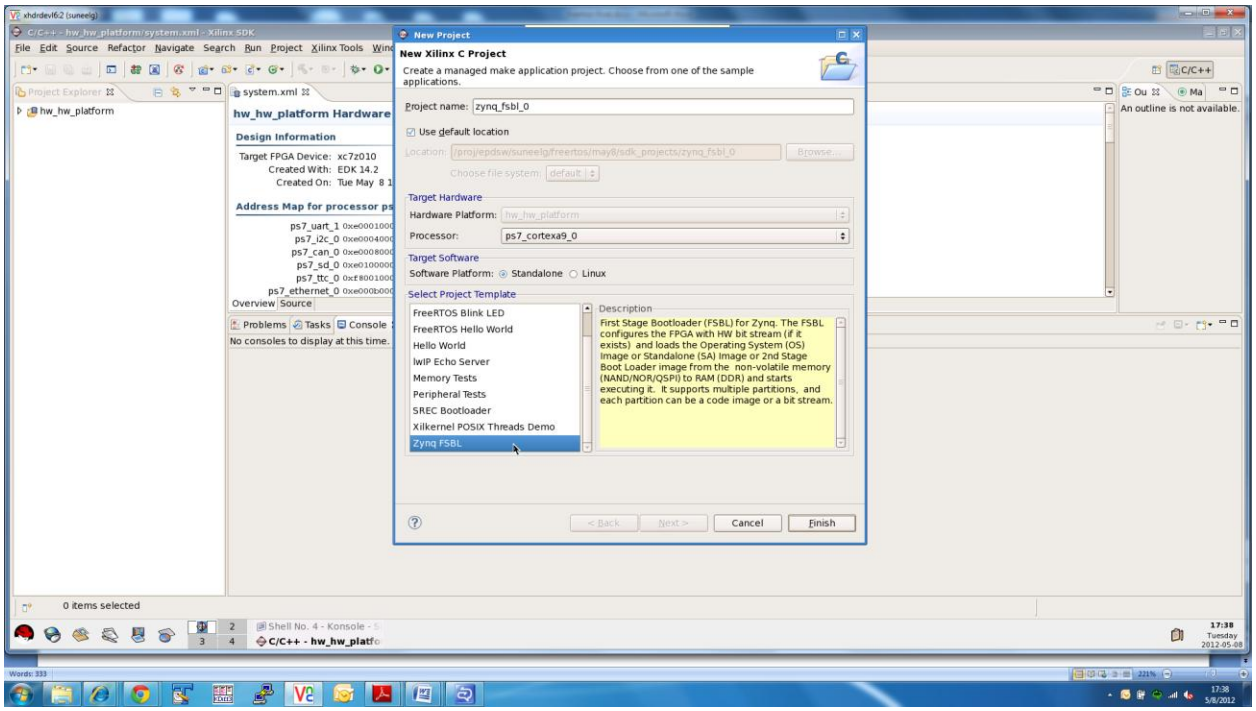



## Create First Stage Bootloader application (FSBL) for Zynq

Before running any application on the Zynq board, a FSBL application must be run to initialize the MIO pins and relevant clocks. To build a FSBL project

File -> New -> Xilinx C Project-> Select.

Under Select Project Template, choose Zynq FSBL and 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” application, the user can see the DS23 LED on ZC702 board blinking.
Similarly 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>/14.1/SDK/SDK/sw/XilinxProcessorIPLib/drivers/scutimer_v1_01_a/src/xscutimer_hw.h
<Install_Dir>/14.1/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