

# RTOS & LwIP on Zynq and Zedboard

**Dr. Heinz Rongen**

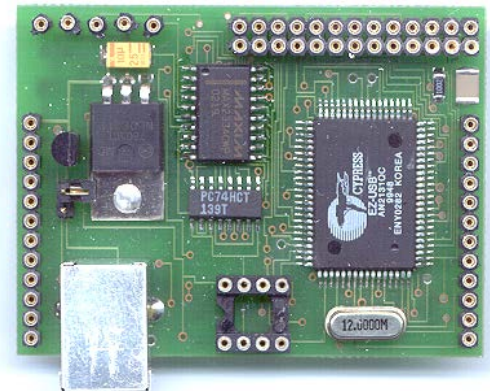
Forschungszentrum Jülich GmbH

Zentralinstitut Systeme der Elektronik (ZEA-2)

[H.Rongen@fz-juelich.de](mailto:H.Rongen@fz-juelich.de)

# µController (The small ones ...)

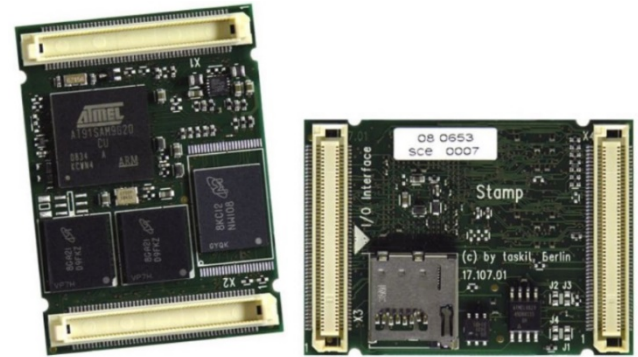
- **Single Chip Solutions / Applications**
- Internal Memory: Flash / RAM (program / data memory)
- Internal I/O
  - Digital I/O: Ports, Timer, Counter, PWM, ...
  - Analog I/O: ADC's, DAC's, ...
  - **Communication: I2C, SPI, RS232, USB, ...**
- **„Bare-Metal“ Programmierung**
  - Without the support of a Operating system
- **Typically: No Networking / Ethernet:**
  - TcpIP Stack to complex to implement



# Big brothers ..

## (Processor Systems)

- **Processor Systems: Multiple Chip Solutions**
  - External Memories
  - External I/O's
- **Operating system support**
  - Windows, embedded Linux, ...
- **Full features Networking / Ethernet:**



Embedded Linux

[www.taskit.de](http://www.taskit.de)

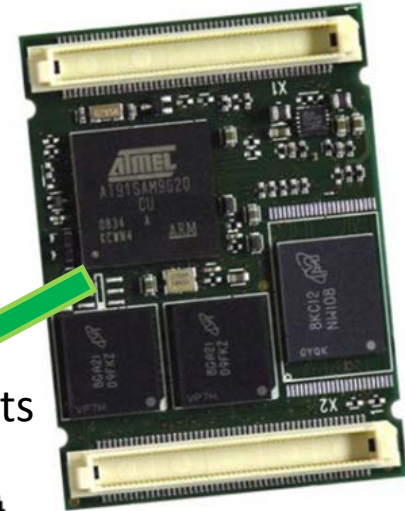
Stamp 926

- ARM9
- 16 MB Flash
- 32 MB SDRAM
- SSD Karte
- LCD/TFT Controller
- **Ethernet**

# The gap ...



- Simple „**single Chip**“ Hardware
- As „Building-Block“ for own HW developments
- **Powerful**
- **OS Support**
  - Multi-Tasking
  - Real-time
- **Ethernet**

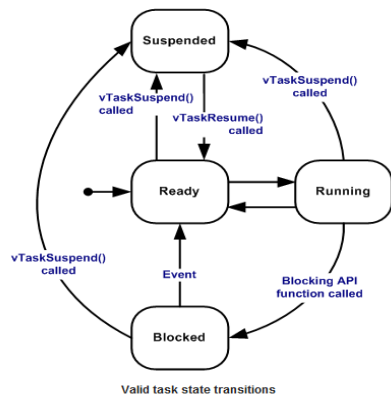


# Solution



[de.wikipedia.org/wiki/FreeRTOS](http://de.wikipedia.org/wiki/FreeRTOS)

- **Open-Source-Echtzeitbetriebssystem**
- **for embedded MicroControllers**
- **Multitasking fähig**
- **präemptive und kooperativer Scheduler**



[lwip.wikia.com/wiki/LwIP\\_Wiki](http://lwip.wikia.com/wiki/LwIP_Wiki)

[IP](#) (Internet Protocol)  
[ICMP](#) (Internet Control Message Protocol)  
[IGMP](#) (Internet Group Management Protocol)  
[UDP](#) (User Datagram Protocol)  
[TCP](#) (Transmission Control Protocol)  
[BSD Berkeley-like socket API](#)  
[DNS](#) (Domain names resolver)  
[SNMP](#) (Simple Network Management Protocol)  
[DHCP](#) (Dynamic Host Configuration Protocol)  
[PPP](#) (Point-to-Point Protocol)  
[ARP](#) (Address Resolution Protocol) for Ethernet

→ in ca. 40 KB code und 8 KB Ram

This on

- Atmel  $\mu$ Controller
- PIC  $\mu$ Controller
- ....
- and ZYNQ

# freeRTOS and LwIP in a FPGA

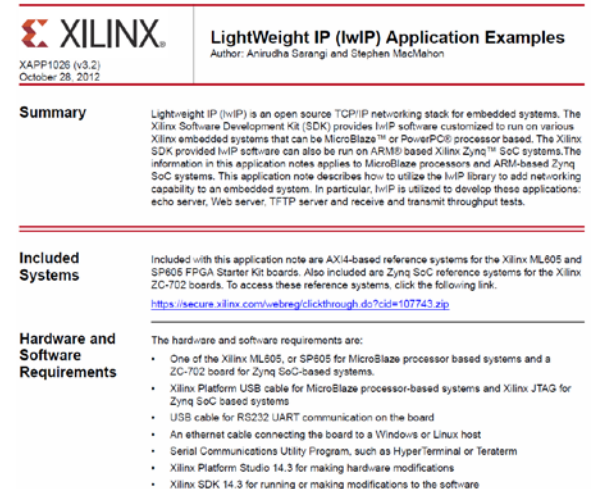
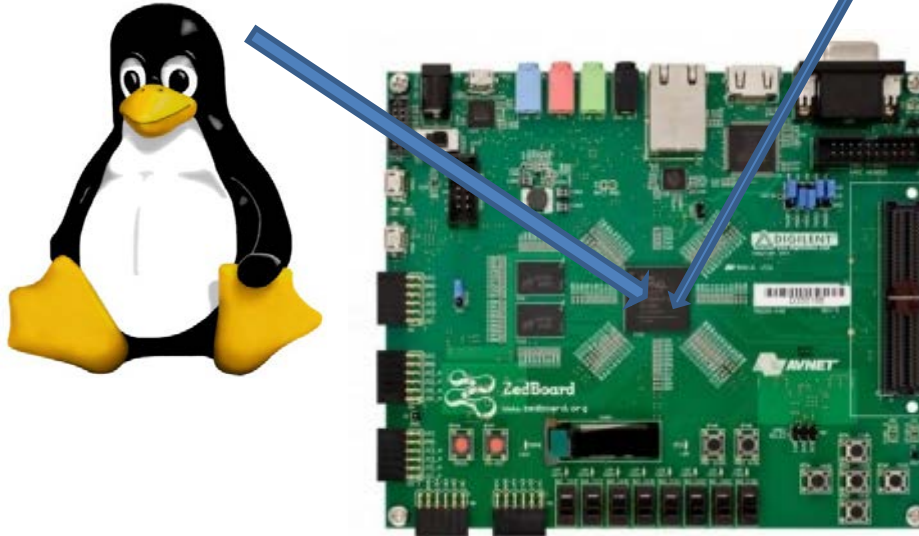
# Xilinx Virtex-7 ZYNQ

- FPGA with internal ARM7 (Cortex-A9)

## → Embedded Linux

## Alternative

- freeRTOS
- lwIP



# freeRTOS and lwIP on Zynq / Zedboard

**Dr. Heinz Rongen**

Forschungszentrum Jülich GmbH

Zentralinstitut Systeme der Elektronik (ZEA-2)

[H.Rongen@fz-juelich.de](mailto:H.Rongen@fz-juelich.de)

## documents

[http://www.xilinx.com/support/documentation/application\\_notes/xapp1026.pdf](http://www.xilinx.com/support/documentation/application_notes/xapp1026.pdf)

- Zedboard example designs: <http://www.zedboard.org/support/design/1521/11>

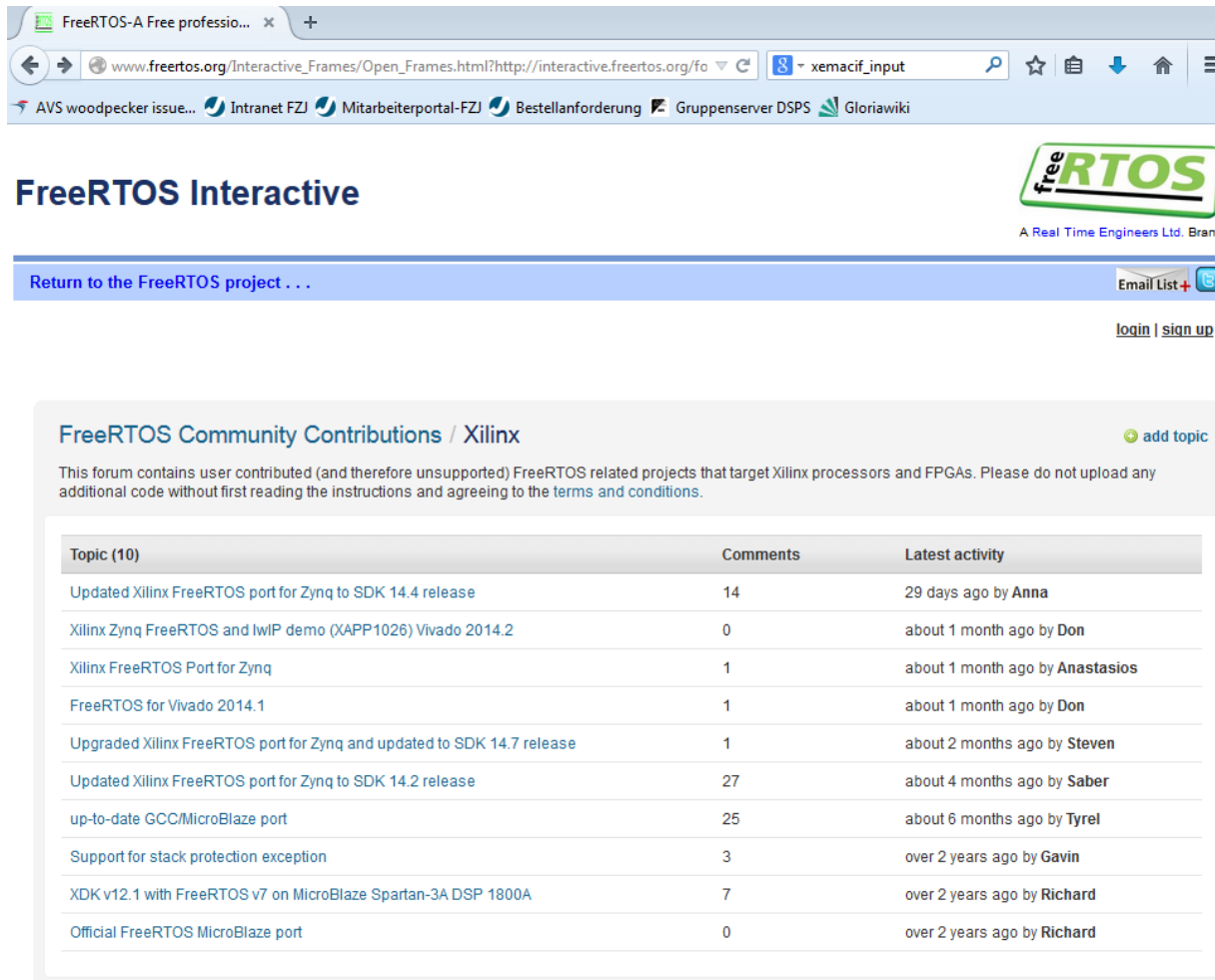
### **ZedBoard version of XAPP1026: LightWeight IP (lwIP) Application Examples**

This ZedBoard adaptation of Xilinx application note XAPP1026 describes how to utilize the lwIP library to add networking capability to an embedded system. In particular, lwIP is utilized to develop these applications: echo server, Web server, TFTP server, as well as receive and transmit throughput tests.



# Download freeRTOS Repository (Library)

- [http://www.freertos.org/Interactive\\_Frames/Open\\_Frames.html?http://interactive.freertos.org/forums](http://www.freertos.org/Interactive_Frames/Open_Frames.html?http://interactive.freertos.org/forums)



The screenshot shows a web browser window with the URL [www.freertos.org/Interactive\\_Frames/Open\\_Frames.html?http://interactive.freertos.org/forums](http://www.freertos.org/Interactive_Frames/Open_Frames.html?http://interactive.freertos.org/forums). The page title is "FreeRTOS Interactive". Below the title, there is a navigation bar with a link to "Return to the FreeRTOS project . . ." and a search bar. The main content area is titled "FreeRTOS Community Contributions / Xilinx" and includes a warning: "This forum contains user contributed (and therefore unsupported) FreeRTOS related projects that target Xilinx processors and FPGAs. Please do not upload any additional code without first reading the instructions and agreeing to the terms and conditions." Below this, there is a table of contributions.

Topic (10)	Comments	Latest activity
<a href="#">Updated Xilinx FreeRTOS port for Zynq to SDK 14.4 release</a>	14	29 days ago by <b>Anna</b>
<a href="#">Xilinx Zynq FreeRTOS and lwIP demo (XAPP1026) Vivado 2014.2</a>	0	about 1 month ago by <b>Don</b>
<a href="#">Xilinx FreeRTOS Port for Zynq</a>	1	about 1 month ago by <b>Anastasios</b>
<a href="#">FreeRTOS for Vivado 2014.1</a>	1	about 1 month ago by <b>Don</b>
<a href="#">Upgraded Xilinx FreeRTOS port for Zynq and updated to SDK 14.7 release</a>	1	about 2 months ago by <b>Steven</b>
<a href="#">Updated Xilinx FreeRTOS port for Zynq to SDK 14.2 release</a>	27	about 4 months ago by <b>Saber</b>
<a href="#">up-to-date GCC/MicroBlaze port</a>	25	about 6 months ago by <b>Tyrel</b>
<a href="#">Support for stack protection exception</a>	3	over 2 years ago by <b>Gavin</b>
<a href="#">XDK v12.1 with FreeRTOS v7 on MicroBlaze Spartan-3A DSP 1800A</a>	7	over 2 years ago by <b>Richard</b>
<a href="#">Official FreeRTOS MicroBlaze port</a>	0	over 2 years ago by <b>Richard</b>

# **Repository (Library) for freeRTOS and lwIP**

## **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 Repository**

- 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.

# Unpack the Repository

The Xilinx Zynq repository in this package has the following structure

**sw**            **Repository used to integrate FreeRTOS related files and related apps in to SDK**

- repo

- - bsp            *FreeRTOS and lwip library Source files*

- - sw\_apps        *Contains Example Apps for Hello World, Blink LED using Semaphore, Blink LED using Mutex , lwip socket, and lwIP raw IO apps*

- The **bsp** folder contains all FreeRTOS and lwip source files. These source files include the generic FreeRTOS source and Zynq related source files (the Portation).
- 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 four LED examples that blink a LEDs. There are also two apps to test the lwIP library, a standalone raw I/O version not using FreeRTOS and a socket I/O version that uses FreeRTOS.

## Create FreeRTOS Application and BSP using Xilinx SDK

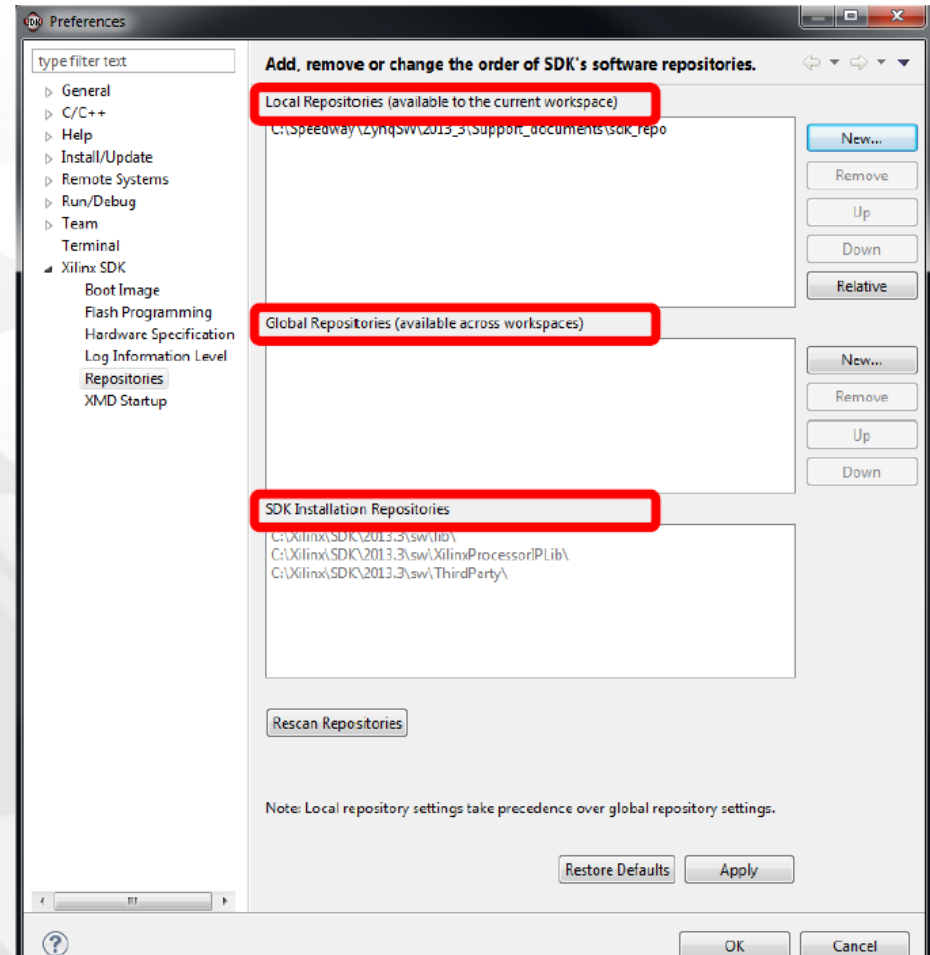
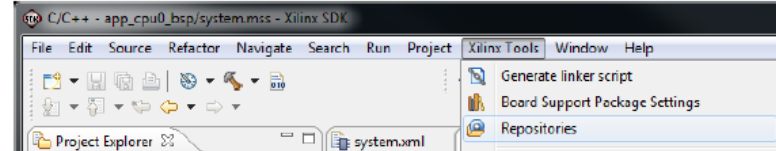
- Extract the zip file available in the package to get the **FreeRTOS\_Zynq\_Vivado** directory.

# Software Repositories

- **May include**

- Custom Drivers
- Custom Libraries
- Custom Board Support Packages
- Could be a modified Xilinx driver or library
- Repository sub-directories
  - bsp
  - drivers
  - sw\_services

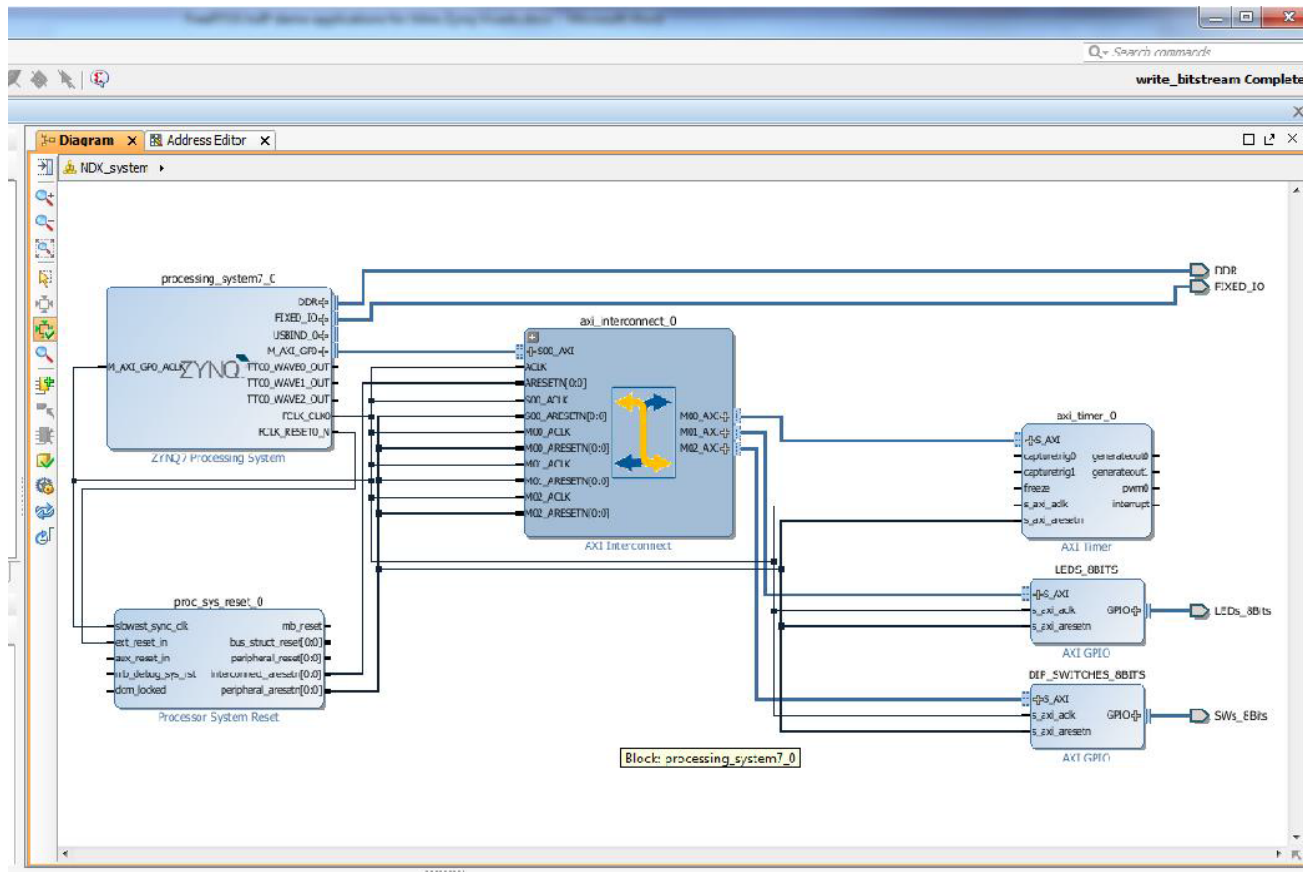
- **Prioritized**



# Create a Zynq PS (Programmable System) for freeRTOS

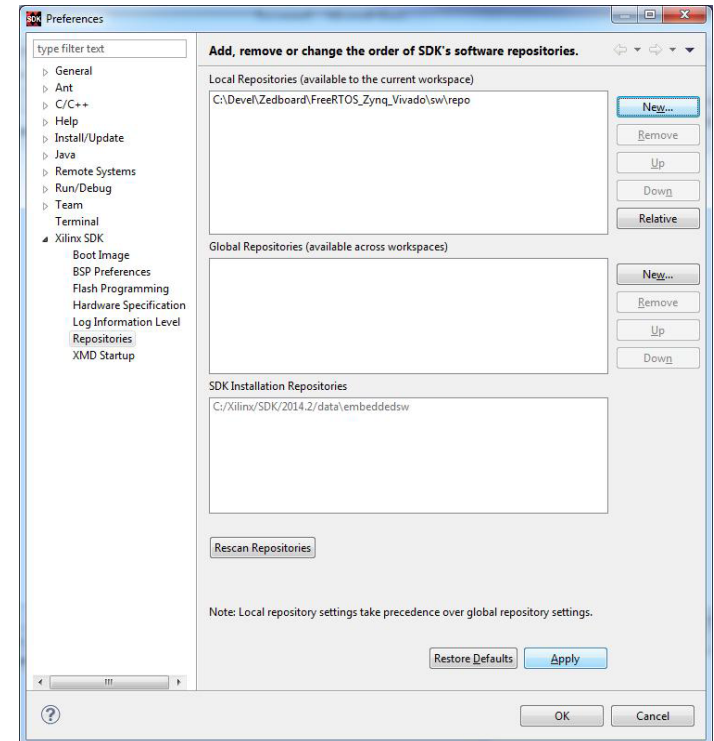
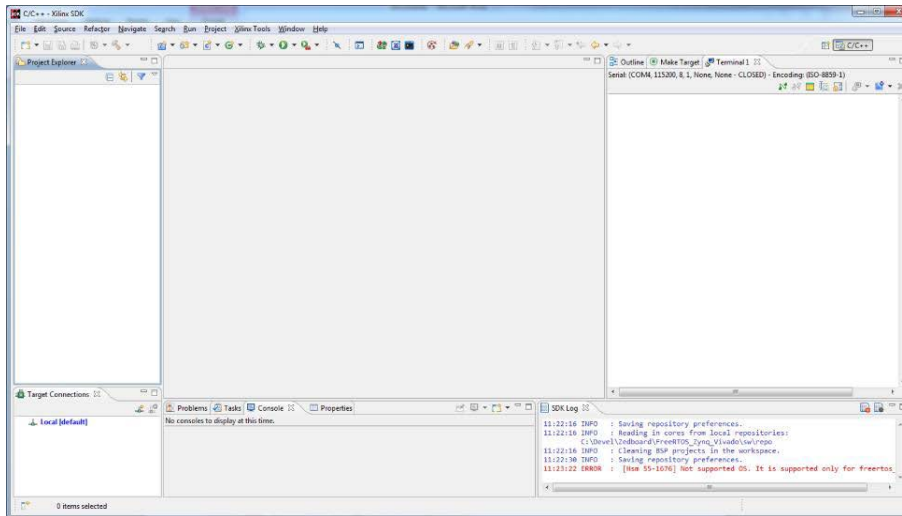
- Configure a Vivado project with the IPI (IP Block Integrator)
- Use the standard MIO Signals (PS Multiplexed IO) to connect GPIO to LEDs
- Use a **AXI\_TIMER** (needed for the RTOS scheduler = Task timing)

An example block design with **AXI GPIO** and **AXI Timer**:



Make a HDL Wrapper, Implement the Design, Generated Bitstream, Export to SDK and launch SDK

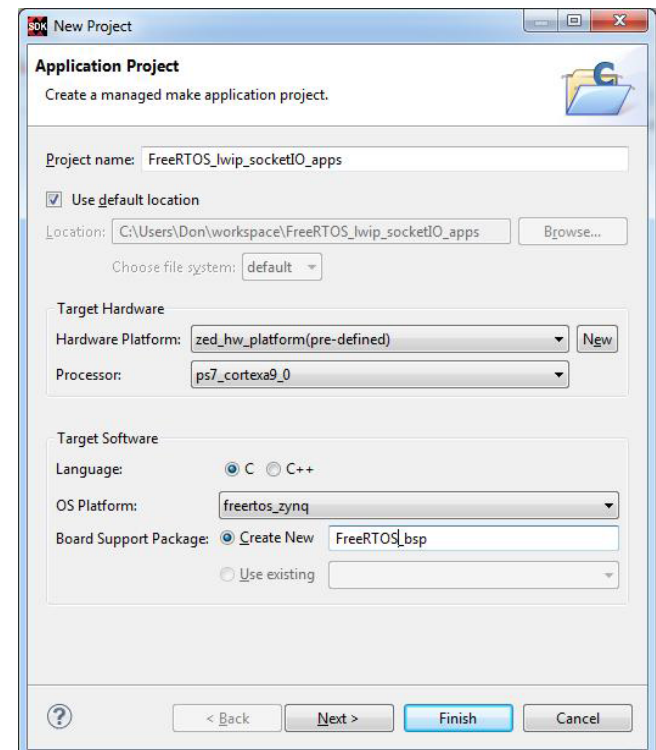
# SDK for Rtos and lwIP Application: Load the Repository



- SDK: Select **Xilinx Tools -> Repositories**.
- Preferences Window pops up.
- Click **New** under *Local Repositories* section and give the path to **FreeRTOS\_Zynq\_Vivado/sw/repo/** directory.
- Click **Rescan** Repositories, then select **Apply** and then **OK**.
- This will ensure that the Xilinx SDK knows about the FreeRTOS and lwIP BSPs and the applications available to it.

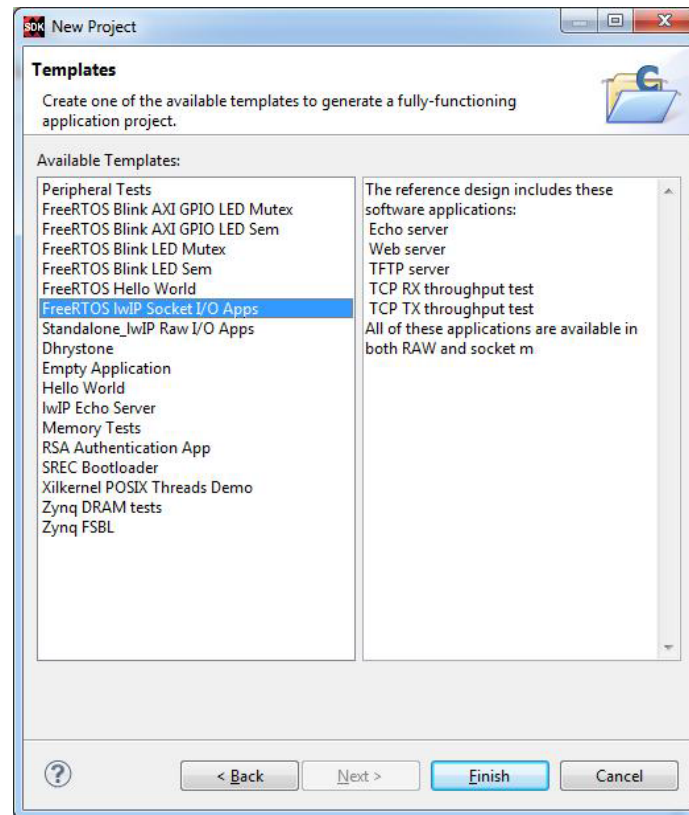
# New Project

- 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 zed\_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. Accept the default or edit and ... **Click Next.**



# Select a Application template

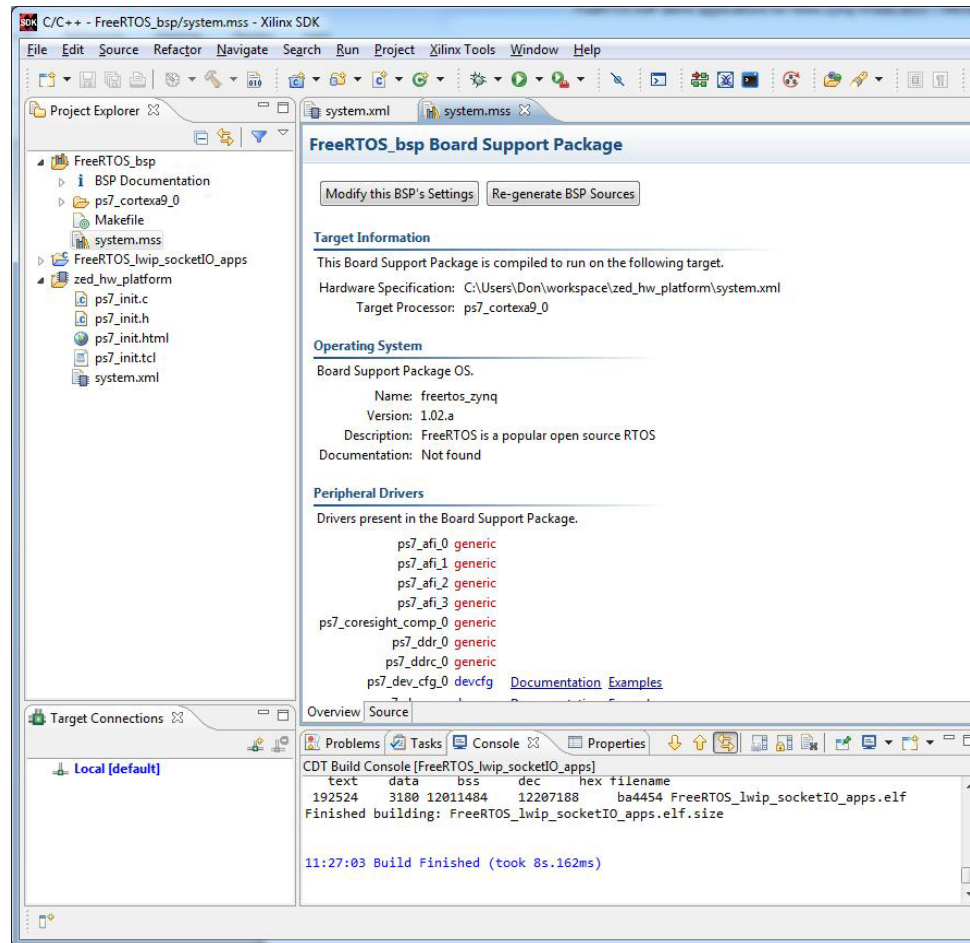
- Select any of the available FreeRTOS applications, such as FreeRTOS lwIP Socket I/O Apps. Click Finish.





# The SDK Project

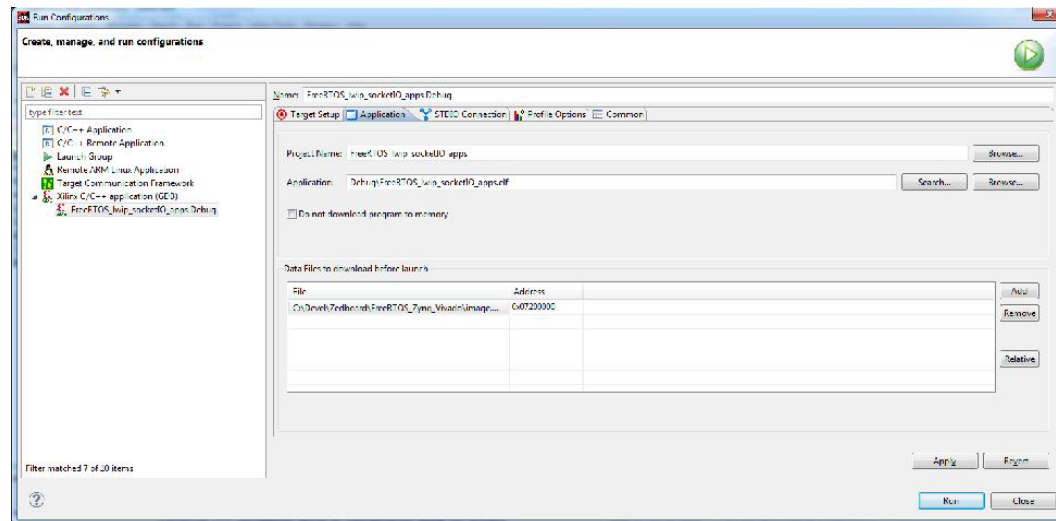
Three sdk projects – FreeRTOS lwIP socket application with the provided project name, board support package and Zedboard hardware platform project will be created.



# Preparing the “Run”

## Execute the lwIP FreeRTOS APPS

- Choose *Run -> Run Configurations ...*
- Under **Xilinx C/ C++** application select **New** from the context menu and choose the lwip Socket app created earlier.
- In the **Application** tab click the “Add” button and browse to the location of the image.mfs file from the XAPP1026 files retrieved from the Xilinx website. Set the address to 0x7200000 and hit “Apply” then “Run”.



- Once this is set up, future launches do not need this dialog. Just run from the run menu directly.
- From a browser on the same subnet navigate to fixed address as reported by the serial terminal output. The terminal will display information about the application.