FREERTOS INSTALLATION INSTRUCTIONS

ECEN 5623 Real Time Embedded Systems

FreeRtos is one of the leading operating systems designed especially for microcontrollers and small microprocessors. FreeRtos is professionally developed to suite the various deployments in the current market. This document contains instructions for porting FreeRtos on Texas Instrumentation Tiva C Series TM4C123G Launchpad and Altera DE1-SOC

Instructions for porting FreeRtos on Tiva C Series TM4C123G Launchpad:

Tiva C Series Launchpad contains ARM cortex M4 as its core. Today we will explore a basic example developed based on FreeRtos. Following are the steps for basic installation of required software for operating Launchpad.

1. Install Code Composer Studio

    Use this if you don't have CCS or have not used Tiva C Series devices before

    a. Download the package for EK-TM4C123GXL-CCS from http://www.ti.com/tool/sw-ek-tm4c123gxl

    b. Find the CSS installer and install it

2. The next important step is to download Tivaware software. It contains all the related examples related to Tiva C Series Launchpad

    a. Download the SW-TM4C package from http://www.ti.com/tool/SW-TM4C or Find the Tivaware installer in the package downloaded from Step 1

3. The final step is to install necessary drivers

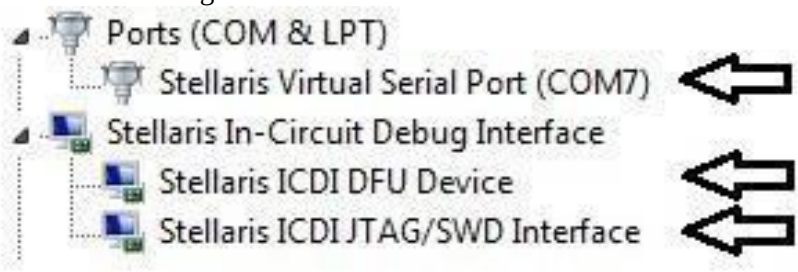
    a. Download the Stellaris ICDI drivers from http://www.ti.com/tool/stellaris_icdi_drivers

    b. Extract the Stellaris ICDI drivers

    c. Plug the LaunchPad into your PC's USB port

    d. Follow the driver installation guide: http://www.ti.com/lit/ml/spmu287c/spmu287c.pdf

    e. Your device manager should show three Stellaris devices like the image below



4. Running example programs

The Tivaware software downloaded in step 2 contains all the necessary examples related to Launchpad. The default path would be C:\ti\TivaWare_C_Series-2.1.3.156\examples\boards\ek-tm4c123gxl\freertos_demo

NOTE: Version of Tivaware may vary in your installation

    a. Select Import from the IDE's **FILE** menu

    b. In the Import dialog box, select "General->Existing Projects into Workspace" then browse to the above directory

    c. Make sure freertos_demo is checked and that Copy Projects into Workspace is not checked before clicking "Finish".

The above folder contains an example project freertos_demo. It makes the LED blinking and allows you to manipulate the colors using buttons (located at the bottom of the board). You just need to connect the board to the PC, open the project (import it) in Code Composer Studio and run it in debug mode. Code Composer Studio will automatically port the program into the board.

For additional information on Queues and semaphores:
http://www.socialledge.com/sjsu/index.php?title=FreeRTOS_Tutorial
Some additional links that might help in installing the necessary software on Launchpad
1. http://www.mcu-turkey.com/wp-content/uploads/2013/12/TM4C123GXL-LaunchPad- Installation-Steps.pdf
2. http://jj09.net/freertos-jump-start/
3. http://www.freertos.org/simple-freertos-demos.html
4. http://processors.wiki.ti.com/index.php/Tiva_TM4C123G_LaunchPad_Blink_the_RGB
5. https://circuitdigest.com/microcontroller-projects/getting-started-with-tiva-c-series-tm4c123g-launchpad-from-texas-instruments

Instructions for porting FreeRtos on Tiva C Series TM4C1294 Launchpad: Getting Started

These instructions will give details on how to create simple template to use FreeRTOS in tm4c1294 connected launchpad

Prerequisites

What software you need to install

1- Download and install the latest version of Code Composer Studio (CCS) and TivaWare as we described here

2-Download and install the latest full version of FreeRTOS source code from here we are using in this project FreeRTOSv10.0.1

Initialize code composer studio

same as we did in test-gpio project

1- create new project
2- build variables for the paths that CCS will need to find your files.
3- Link driverlib.lib to Your Project
4- Add the INCLUDE search paths for the header files

Include FreeRTOS files into project

1- click right mouse button on project folder and select new->Folder and name it Source.
2- import to that created Source folder all files that are in Source directory from downloaded FreeRTOS package. here We have added only kernel C files
3- To include port files create folder named portable inside Source folder. And then in portable folder we create CCS folder. And in CCS we create folder named ARM_CM4F. Then import porting files port.c,portmacro.h and portasm.asm to this folder from FreeRTOS package.
4- to include memory management file heap_1.c, which takes care of allocating and freeing memory for tasks and queues. create MemMang folder in portable folder and add file from same folder in downloaded package
5- And lastly FreeRTOS needs FreeRTOSConfig.h configuration file that keeps all freeRTOS related settings. Just import it from C:\ti\TivaWare_C_Series-2.1.4.178\examples\boards\ek-tm4c123gxl\freertos_demo.
6- create another folder Drivers in project root directory. This will be used to store microcontroller peripheral drivers like USART, I2C, ADC, button, LED and so on.

7- configure the project by going to project then select properties
8- include all directories containing .h files in your project by right click on project name --> properties --> include options then add all folders of free rtos containing .h files
9- in start up file add the following functions

- extern void xPortPendSVHandler(void);
- extern void vPortSVCHandler(void);
- extern void xPortSysTickHandler(void);

10- in start up file add xPortPendSVHandler, vPortSVCHandler, xPortSysTickHandler for the interrupt handllers (The PendSV handler ,SVCall handler ,The SysTick handler)

Instructions for porting FreeRtos on Altera DE1-SOC:

FreeRtos can be directly ported using EDS tools for ARM or using Linux installed. For better reliability it is recommended to import FreeRtos using DS-5 Eclipse environment. However interested students can explore porting FreeRtos on to Linux environment.

1. Download EDS tools form: http://www.freertos.org/RTOS_Altera_SoC_ARM_Cortex-A9.html
2. Download FreeRtos Source Code examples: http://www.freertos.org/a00104.html
3. Make sure to install GCC cross compiler tools
4. The ARM DS-5 Eclipse project file is located in the

FreeRTOS/Demo/CORTEX_A9_Cyclone_V_SoC_DK directory

5. Select Import from the IDE's **FILE** menu
6. In the Import dialog box, select "General->Existing Projects into Workspace" then browse to and select the FreeRTOS/Demo/CORTEX_A9_Cyclone_V_SoC_DK directory. A project called "RTOSDemo" will be visible.
7. Make sure RTOSDemo is checked and that Copy Projects into Workspace is not checked before clicking "Finish".
8. It is necessary to update the project to set the GCC compiler's location.
9. Select "Properties" from the IDE's "Project" menu to bring up the properties dialogue box. In the dialogue box select "C/C++Build->Settings-> Cross Settings", then set the path to the compiler to be correct for your installation.

Note: If the dialogue box tab shown is missing, or contains an error message, then it is likely your version of DS-5 does not have the CDT cross compiler plug in installed. If this is the case the plug in can be installed manually by following the instructions provided on the ARM website (https://developer.arm.com/products/software-development-tools/ds-5-development-studio/resources/tutorials/adding-new-compiler-toolchains-to-ds-5). It will be necessary to create a new workspace after the plug-in has been installed.

10. Open main.c and set mainCREATE_SIMPLE_BLINKY_DEMO_ONLY to generate either the simple blinky demo, or the full test and demo application, as required.
11. Select "Build All" from the IDE's "Project" menu.
12. After the build completes, select "Debug Configurations..." from the IDE's "Debug" menu, and configure and run a debug configuration that is appropriate for your selected connection method

Instructions for porting FreeRTOS on Windows:

http://www.freertos.org/FreeRTOS-Windows-Simulator-Emulator-for-Visual-Studio-and-Eclipse-MingW.html

Some Important links:

- The following link contains porting instructions for Microsoft Visual studio: It also demonstrates main functionality of CREATE_SIMPLE_BLINKY_DEMO_ONLY example
http://www.freertos.org/simple-freertos-demos.html

- The following link contains porting instructions for Eclipse/ MinGW
http://www.freertos.org/FreeRTOS-Windows-Simulator-Emulator-for-Visual-Studio-and-Eclipse MingW.html

- http://www.freertos.org/RTOS_Altera_SoC_ARM_Cortex-A9.html