CC32xx TI-RTOS

Return to CC31xx & CC32xx Home Page

Return to CC31xx Sample Applications

Introduction

The CC32xx SDK includes a prebuilt TI-RTOS Kernel (formerly known as SYS/BIOS) and has some examples showcasing the usage of the applications with TI-RTOS kernel. The SDK includes an OS abstraction layer that is also ported for TI-RTOS. The SDK also includes a power management framework to aid the exercising of the low power modes in the device. There are also high level drivers that interface with the power management framework.

It is recommended to use the CC32xx SDK for applications requiring the flexibility to easily migrate to any RTOS and also if the usecase exercises the low power modes of the device.

"TI-RTOS for SimpleLink Wireless MCUs" package exercises more features of TI-RTOS (TI-RTOS Kernel, TI-RTOS Drivers and Board Initialization, ...) and has example applications showcasing various usecases. TI-RTOS Drivers and Board Initialization provides a set of device driver APIs, that are standard across all devices, as well as initialization code for all supported boards.

It is recommended to use the "TI-RTOS for SimpleLink Wireless MCUs" package to exploit all the features supported by TI-RTOS. Currently, in order to support exercising the low power modes of the device, the power management framework has to be imported and adapted from the CC32xx SDK package. Built-in support to exercise low power modes in this package is planned to be available in future releases.

Using TI-RTOS with CC32xx SDK

TI-RTOS for CC3xxx 2.01.00.03 has been used with CC3200 applications.

It can be installed from the link from the app center of CCS v6.0.1

SDK PreBuilt Binaries

The prebuilt binaries that are part of the SDK are built with **TI-RTOS for CC3xxx 2.01.00.03** version of the TI-RTOS components.
Any of the following tool set can been used(for building the application binaries and the ti_rtos library) for TI-RTOS on CC3200 platform:
1. CCS version: CCSv6.0.1
2. IAR version: ARM v7.2
3. GCC

In case any of these tools need to be downloaded, use the following paths:
CCS:
http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v6 [1]
IAR:
http://www.iar.com/Service-Center/Downloads/ [2]

Steps to build TI-RTOS configuration project:

1. CCS:

a) Import the ti_rtos_config project into CCS (ti_rtos_config\ccs folder).

b) Right click and select the 'Rebuild All' option.

c) To change the version of TI-RTOS, the RTSC tab in the Build options needs to be used.

2. IAR:

a) Run the 'ti_rtos_config\ewarm\buildos-iar.bat' batch file to rebuild the configuration project for IAR.

b) The 'CGTOOLS' macro in the batch file refers to the IAR installation path in the PC.

c) The 'XDCTOOLS' macro is used to set the XDC tool set path used by TI-RTOS for building.

d) The 'SYSBIOS' macro is used to set the TI-RTOS SYS/BIOS installation path.

e) The above paths need to be modified to build a new TI-RTOS version.

3. GCC:

a) The make file 'ti_rts_config\gcc\make' is to be used to rebuild the ti_rtos config project for GCC.

b) The 'tools\gcc_scripts\makedefs_ti_rtos' file defines the build macros.

c) The 'XDCTOOLS_INSTALLATION_DIR' macro is used to set the XDC tool set path used by TI-RTOS for building.

d) The 'TIRTOS_INSTALLATION_DIR' macro sets the TI-RTOS installation directory.

e) The 'BIOS_INSTALLATION_DIR' macro sets the SYS/BIOS installation directory for TI-RTOS.

f) The above macros have to be updated to build the config project for a new TI-RTOS version.

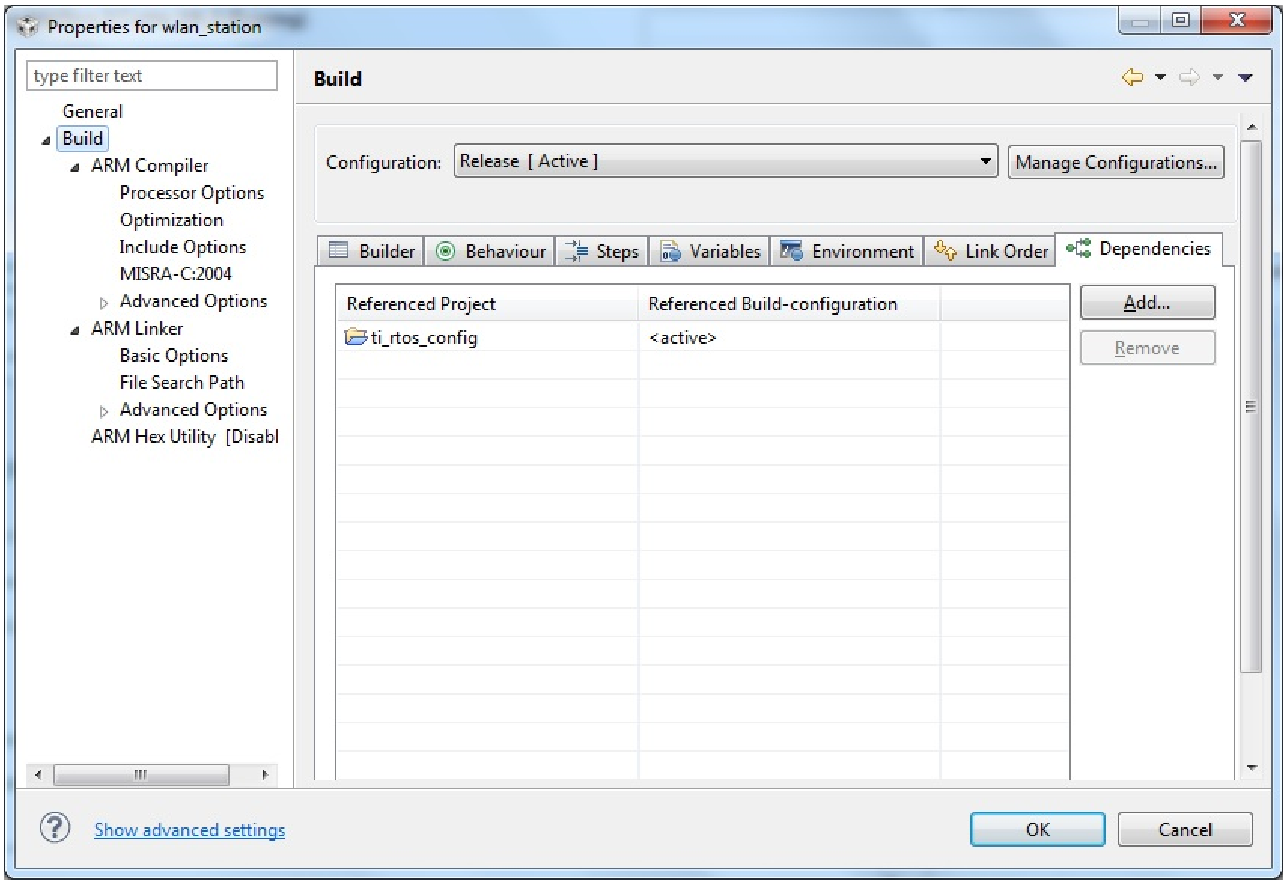
NOTE:

The 'Info Center-Get Time Application' application demonstrates building an application for TI-RTOS on IAR, GCC and CCS.

Steps to build a new application on CCS with TI-RTOS:

1. **Import TI-RTOS project** - For the application to work with TI-RTOS, **ti_rtos_config** project needs to be imported into the application workspace. This projects can be found in CC3200-SDK under **ti_rtos** folder.

2. **Add TI-RTOS as dependency** - The application should add ti_rtos_config as dependency to enable the RTSC functionality in the project.



Configuration: Release [Active]

Referenced Project	Referenced Build-configuration
ti_rtos_config	<active>

3. **Including TI-RTOS library -** oslib project contains the OS adaptation layer files giving general wrapper functions around TI-RTOS function calls. Building this project with the ti_rtos configuration gives a ti_rtos.a library which should be included in the project for using RTOS functionality in the project.

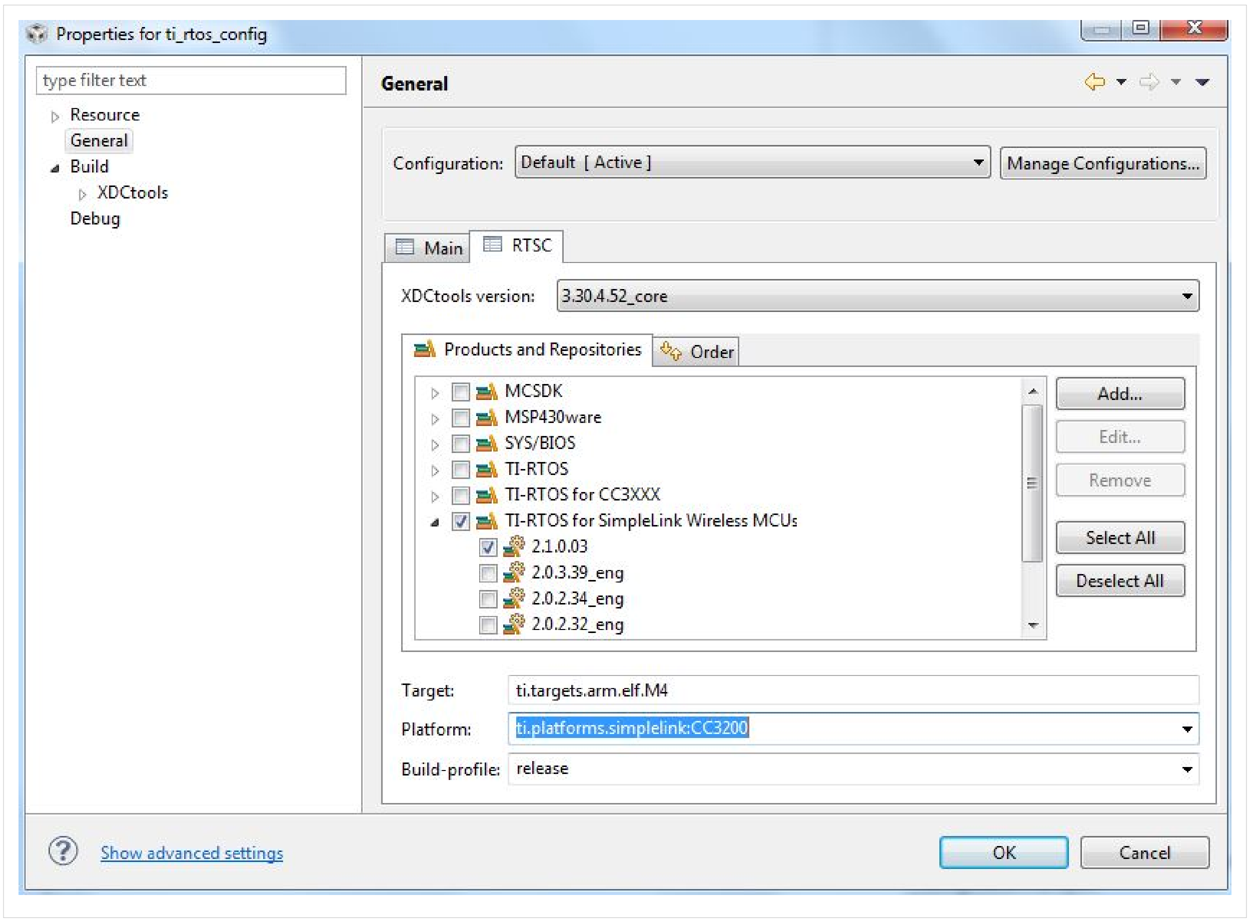
4. **Registering Interrupts(if any) -** All interrupts have to be registered to the NVIC using the osi_IntRegister() call.(Please refer osi.c for details). This creates a Hwi in TI-RTOS for a hardware interrupt.

Note:

App.cfg present in the ti_rtos_config contains a common configuration files. The benefit of having a separate config project is that several applications can use the same configuration. This helps save on build times and allows a team to share a common configuration set up by a system integrator.

Notes:

1. For building any of the TI-RTOS based applications, the ti_rtos_config project, which is a dependency project, has to be imported in the workspace.
2. If choosing the RTSC platform in the ti_rtos_config package, the platform name has to *ti.platforms.simplelink:CC3200*.



General

Configuration: Default [Active] Manage Configurations...

XDCtools version: 3.30.4.52_core

- [ ] MCSDK
- [ ] MSP430ware
- [ ] SYS/BIOS
- [ ] TI-RTOS
- [ ] TI-RTOS for CC3XXX
- [✓] TI-RTOS for SimpleLink Wireless MCUs
  - [✓] 2.1.0.03
  - [ ] 2.0.3.39_eng
  - [ ] 2.0.2.34_eng
  - [ ] 2.0.2.32_eng

Target: ti.targets.arm.elf.M4

Platform: ti.platforms.simplelink:CC3200

Build-profile: release

1. For hooking a function to the idle task(as in the examples using Power Management Framework), the following lines have to be added in app.cfg file in ti_rtos_config project.

```
var Idle = xdc.useModule('ti.sysbios.knl.Idle');
Idle.addFunc('&IdleHookFunction');
```

where **IdleHookFunction()** is the function that needs to be hooked to the idle task and may vary depending upon the application.

Archives

The latest software packages are the ones which are currently supported.

References

[1] http://processors.wiki.ti.com/index.php/Category:Code_Composer_Studio_v6
[2] http://www.iar.com/Service-Center/Downloads/

Article Sources and Contributors

CC32xx TI-RTOS *Source*: http://processors.wiki.ti.com/index.php?oldid=185069 *Contributors*: Jitgupta

Image Sources, Licenses and Contributors

File:Cc31xx cc32xx return home.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Cc31xx_cc32xx_return_home.png *License*: unknown *Contributors*: A0221015

File:Cc32xx return sample apps.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Cc32xx_return_sample_apps.png *License*: unknown *Contributors*: A0221015

Image:Dependency2.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:Dependency2.jpg *License*: unknown *Contributors*: Jitgupta

Image:TI-RTOS_2_1_0_3.JPG *Source*: http://processors.wiki.ti.com/index.php?title=File:TI-RTOS_2_1_0_3.JPG *License*: unknown *Contributors*: Jitgupta