TI-RTOS 2.14 for MSP43x

Getting Started Guide



Literature Number: SPRUHU4C October 2015



Contents

Pre	eface	
1	Abo	at TI-RTOS
-	1.1	What is TI-RTOS?
	1.2	What are the TI-RTOS Components?5
	1.3	How Can I Find Example Projects?
	1.4	What Compilers and Targets are Supported?
	1.5	What Boards and Devices Have TI-RTOS Driver Examples?
	1.6	What Drivers Does TI-RTOS Include?
	1.7	For More Information
_	l t	III. TI DTOC
2		Illing TI-RTOS
	2.1	System Requirements
	2.2	Installing Code Composer Studio
	2.3	Installing TI-RTOS in Code Composer Studio
	2.4	Installing TI-RTOS for Use in IAR Embedded Workbench
	2.5	Installing TI-RTOS as a Standalone Product
3	Exar	nples for TI-RTOS13
	3.1	Creating Example Projects Using the Resource Explorer in CCS
		3.1.1 Creating an Empty TI-RTOS Project
		3.1.2 Creating Examples to Build via a Command Line
	3.2	Creating Examples with IAR Embedded Workbench
	3.3	Summary of Example Peripheral Use and Target Support
		3.3.1 Readme Files and Common Driver Example Features
	3.4	MSP430 MSP-EXP430F5529LP Settings and Resources
	3.5	MSP430 MSP-EXP430FR5969LP Settings and Resources
	3.6	MSP430 MSP-EXP430FR6989LP Settings and Resources
	3.7	MSP432 MSP-EXP432P401RLP Settings and Resources
	3.8	BoosterPacks
		3.8.1 SD Card BoosterPack
		3.8.2 TMP006 BoosterPack
		3.8.3 Sharp Memory LCD BoosterPack
		3.8.4 CC3100 BoosterPacks
		3.8.5 RF430CL330 NFC Transponder Module
		3.8.6 TPL0401EVM Board
	3.9	Installing USB Drivers for the USB Device Examples
4	Conf	iguring TI-RTOS
	4.1	Starting the Configuration Tool
	4.2	Configuring TI-RTOS Drivers
		4.2.1 Configuring System Support
	4.3	Configuring Components of TI-RTOS
Ind		
1110	ICA .	



Read This First

About This Manual

This manual describes TI-RTOS for MSP43x. The version number as of the publication of this manual is v2.14.

Notational Conventions

This document uses the following conventions:

• Program listings, program examples, and interactive displays are shown in a special typeface. Examples use a bold version of the special typeface for emphasis.

Here is a sample program listing:

```
#include <xdc/runtime/System.h>
int main(void)
{
    System_printf("Hello World!\n");
    return (0);
}
```

Square brackets ([and]) identify an optional parameter. If you use an optional parameter, you
specify the information within the brackets. Unless the square brackets are in a **bold** typeface, do not
enter the brackets themselves.

Trademarks

Registered trademarks of Texas Instruments include Stellaris and StellarisWare.

Trademarks of Texas Instruments include: the Texas Instruments logo, Texas Instruments, TI, TI.COM, C2000, C5000, C6000, Code Composer, Code Composer Studio, Concerto, controlSUITE, DSP/BIOS, MSP430, MSP430Ware, MSP432, SimpleLink, Sitara, SPOX, TI-RTOS, Tiva, TivaWare, TMS320, TMS320C5000, TMS320C6000, and TMS320C2000.

ARM is a registered trademark, and Cortex is a trademark of ARM Limited.

Windows is a registered trademark of Microsoft Corporation.

Linux is a registered trademark of Linus Torvalds.

IAR Systems and IAR Embedded Workbench are registered trademarks of IAR Systems AB.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

October 5, 2015



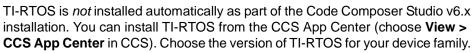
About TI-RTOS

This chapter provides an overview of TI-RTOS for MSP43x.

Topic	Page
1.1 What is TI-RTOS?	4
1.2 What are the TI-RTOS Components?	5
1.3 How Can I Find Example Projects?	6
1.4 What Compilers and Targets are Supported?	7
1.5 What Boards and Devices Have TI-RTOS Driver Examples?	7
1.6 What Drivers Does TI-RTOS Include?	7
1.7 For More Information	8

1.1 What is TI-RTOS?

TI-RTOS is a scalable, one-stop embedded tools ecosystem for TI devices. It scales from a real-time multitasking kernel (SYS/BIOS) to a complete RTOS solution including additional middleware components and device drivers. By providing essential system software components that are pre-tested and pre-integrated, TI-RTOS enables you to focus on creating your application.





If you use devices in multiple families, you can install multiple TI-RTOS versions. See Section 2.3 for details.

If you do not use CCS, you can download and install TI-RTOS as a standalone product (see Section 2.5). In addition to the Texas Instruments Code Generation Tools, TI-RTOS includes support for the IAR tool chain (see Section 2.4).

TI-RTOS is provided with full source code and requires no up-front or runtime license fees.



1.2 What are the TI-RTOS Components?

TI-RTOS contains its own source files, pre-compiled libraries (both instrumented and non-instrumented), and examples. Additionally, TI-RTOS contains a number of components within its "products" subdirectory. The components of TI-RTOS for MSP43x are as follows.

Table 1-1. TI-RTOS Components

TI-RTOS Component	Name	PDF Documentation Location
TI-RTOS	TI-RTOS examples	Chapter 3 of this Getting Started Guide
TI-RTOS Kernel	SYS/BIOS	SYS/BIOS (TI-RTOS Kernel) User's Guide SPRUEX3
TI-RTOS Instrumentation	UIA	System Analyzer User's Guide SPRUH43
TI-RTOS File System	FatFS	TI-RTOS User's Guide SPRUHD4
TI-RTOS USB	USB stack	TI-RTOS User's Guide SPRUHD4
TI-RTOS Drivers and Board Initialization	Drivers and MSPWare	TI-RTOS User's Guide SPRUHD4

The components in the "products" subdirectory are:

- TI-RTOS Kernel SYS/BIOS. SYS/BIOS is a scalable real-time kernel. It is designed to be used
 by applications that require real-time scheduling and synchronization or real-time instrumentation. It
 provides preemptive multi-threading, hardware abstraction, real-time analysis, and configuration
 tools. SYS/BIOS is designed to minimize memory and CPU requirements on the target.
- **TI-RTOS Instrumentation UIA.** The Unified Instrumentation Architecture (UIA) provides target content that aids in the creation and gathering of instrumentation data (for example, Log data).
- MSPWare. This software is an extensive suite of drivers, code examples, and design resources
 designed to simplify and speed development of MSP43x microcontroller applications. This
 component is customized and built as a library to support MSP430F5xx_6xx devices and MSP432
 devices with TI-RTOS.¹
- XDCtools. This core component provides the underlying tooling for configuring and building TI-RTOS and its components. XDCtools is installed as part of CCS v6.x. If you install TI-RTOS outside CCS, a compatible version of XDCtools is installed automatically. XDCtools is installed in a directory at the same level as TI-RTOS, not in the "products" directory of the TI-RTOS installation.

^{1.} Customization details for MSPWare and instructions for building MSPWare's driverlib libraries for MSP430F5xx6xx families are provided in the TI-RTOS.README in the top-level directory of the MSPWare component.

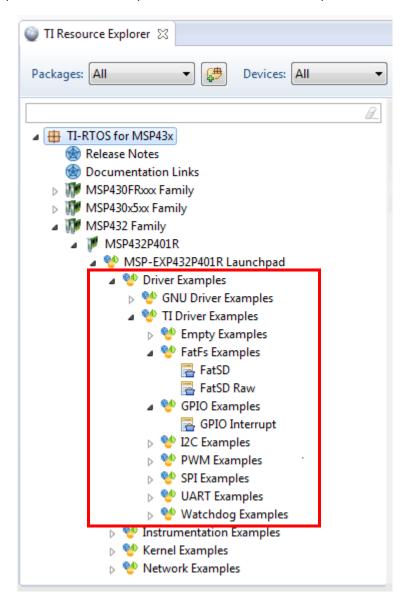


1.3 How Can I Find Example Projects?

TI-RTOS and its components provide numerous examples that you can import using the **Resource Explorer** in Code Composer Studio (CCS). These examples use TI-RTOS and its components and have all the settings needed for your device. Expand the tree in the Resource Explorer to see the examples that are available for your device. Resource Explorer provides TI-RTOS examples for both the TI and GNU tool chains.

- Driver Examples are TI-RTOS driver examples.
- Instrumentation Examples are UIA examples.
- Kernel Examples are the SYS/BIOS examples.
- Network Examples show how to use capabilities such as WiFi, PPP, Ethernet, and HTTP/HTTPS.

Follow the steps in Section 3.1 to import, build, and run these examples.





1.4 What Compilers and Targets are Supported?

The following code generation tool (compilers and linkers) used to build the TI-RTOS libraries and to perform testing. More recent versions are expected to be compatible.

- Texas Instruments: MSP430 CodeGen Tools 4.4.2 (for MSP430)
- Texas Instruments: ARM CodeGen Tools 5.2.2 (for MSP432)
- IAR Workbench for MSP430: 5.60.1 (for MSP430)
- IAR Workbench for ARM: 7.40.1 (for MSP432)

The configuration uses a "target" specification during the build. This specification is sometimes called the "RTSC target." The targets supported are:

- ti.targets.msp430.elf.MSP430X (for MSP430)
- iar.targets.msp430.MSP430X small (for MSP430)
- ti.targets.arm.elf.M4F (for MSP432)

1.5 What Boards and Devices Have TI-RTOS Driver Examples?

Currently, TI-RTOS provides driver examples for the following boards:

Family	Device on Board	Board
MSP430F5xx	MSP430F5529	MSP-EXP430F5529LP LaunchPad
MSP430FR5xx	MSP430FR5969	MSP-EXP430FR5969LP LaunchPad
MSP430FR6xx	MSP430FR6989	MSP-EXP430FR6989LP LaunchPad
MSP432	MSP432P401R	MSP-EXP432P401RLP LaunchPad

Examples are provided specifically for the supported boards, but libraries are provided for each of these device families, so that you can port the examples to similar boards. Porting information for TI-RTOS is provided on the Texas Instruments Wiki.

For MSP43x devices, TI-RTOS drivers are prebuilt only for MSP430F5529, MSP430FR5969, and MSP432P401R devices. TI-RTOS drivers for other MSP43x devices must be added into the TI-RTOS build system. See "Rebuilding MSPWare's driverlib for TI-RTOS and its Drivers" in the TI-RTOS User's Guide (SPRUHD4) for details.

1.6 What Drivers Does TI-RTOS Include?

TI-RTOS includes drivers for the following peripherals. These drivers are in the <install_dir>/packages/ti/drivers directory. TI-RTOS examples show how to use these drivers. Note that all of these drivers are built on top of MSPWare.

The TI-RTOS installation installed drivers for multiple device families. The following list indicates which drivers can be use with MSP43x targets:

- I²C. API set intended to be used directly by the application or middleware.
- **GPIO.** API set intended to be used directly by the application or middleware to manage the GPIO interrupts, pins, and ports.
- **SPI.** API set intended to be used directly by the application or middleware to communicate with the Serial Peripheral Interface (SPI) bus. SPI is sometimes called SSI (Synchronous Serial Interface).



For More Information www.ti.com

- **SDSPI.** Driver for SD cards using a SPI (SSI) bus. This driver is used by the FatFS and not intended to be called directly by the application.
- UART. API set intended to be used directly by the application to communicate with the UART.
- Other USB functionality. See the USB examples for reference modules that provide support for the Human Interface Device (HID) class (mouse and keyboard) and Communications Device Class (CDC). This code is provided as part of the examples, not as a separate driver.
- Watchdog. API set intended to be used directly by the application or middleware to manage the watchdog timer.
- **WiFi.** Driver used by a Wi-Fi device's host driver to exchange commands, data, and events between the host MCU and the wireless network processor. Not intended to be called directly.

1.7 For More Information

To see release notes for a component, go to the subdirectory for that component within the TI-RTOS products directory. For example, C:\ti\tirtos_msp43x_2_##_##_##\products\bios_6_40_##_## contains release notes for SYS/BIOS.

To see user guide PDFs and other documentation for a component, go to the "docs" subdirectory within the directory that contains the release notes.

To learn more about TI-RTOS and its components, refer to the following documentation:

TI-RTOS

- TI-RTOS User's Guide (SPRUHD4)
- TI-RTOS on the Texas Instruments Wiki
- TI-RTOS forum on TI's E2E Community
- TI-RTOS Porting Guide
- Embedded Software Download Page

Code Composer Studio (CCS)

- CCS online help
- CCSv6 on the Texas Instruments Wiki
- Code Composer forum on TI's E2E Community

SYS/BIOS

- SYS/BIOS User's Guide (SPRUEX3)
- SYS/BIOS API and configuration reference. In Resource Explorer (in CCS), choose the
 Documentation Links item for your version of TI-RTOS. In the Documentation Links page,
 choose the TI-RTOS Kernel Runtime APIs and Configuration (cdoc) item.
- SYS/BIOS on the Texas Instruments Wiki
- TI-RTOS forum on TI's E2E Community
- SYS/BIOS 6.x Product Folder

XDCtools

- SYS/BIOS User's Guide (SPRUEX3)
- XDCtools online reference. Open from CCS help or run <xdc_install>/docs/xdctools.chm.
- RTSC-Pedia Wiki
- TI-RTOS forum on TI's E2E Community

www.ti.com For More Information

UIA

- System Analyzer User's Guide (SPRUH43)
- UIA API and configuration reference. In Resource Explorer (in CCS), choose the
 Documentation Links item for your version of TI-RTOS. In the Documentation Links page,
 choose the TI-RTOS Instrumentation Runtime APIs and Configuration (cdoc) item.
- System Analyzer on the Texas Instruments Wiki

MSPWare

- Documents in <tirtos_install>/products/MSPWare_##/doc
- MSPWare Product Folder

FatFS API

- Open source documentation
- FatFS for SYS/BIOS wiki page
- SYS/BIOS API and configuration reference. In Resource Explorer (in CCS), choose the
 Documentation Links item for your version of TI-RTOS. In the Documentation Links page,
 choose the TI-RTOS Kernel Runtime APIs and Configuration (cdoc) item and see help under
 the ti.sysbios.fatfs.FatFS module topic.

General microcontroller information

Microcontrollers forum on TI's E2E Community

MSP43x boards and devices

- MSP-EXP430F5529LP LaunchPad Evaluation Kit
- MSP430 USB Developers Package (which includes the USB Descriptor Tool)
- MSP430F5529 Microcontroller
- MSP-EXP430FR5969LP LaunchPad Evaluation Kit
- MSP430FR6989 Microcontroller
- MSP-EXP430FR6989LP LaunchPad Evaluation Kit

BoosterPacks

- EM Adapter BoosterPack
- RF430CL330H NFC Dynamic Tag Target Board
- MicroSD BoosterPack
- SD Card BoosterPack
- TMP006 BoosterPack
- I2C TPL0401EVM BoosterPack

SD Cards

Specification

I²C

Specification

WiFi

- SimpleLink Wi-Fi CC31xx/CC32xx Wiki Page
- CC3100 Product Folder
- CC3200 Product Folder
- SimpleLink Wi-Fi SmartConfig Apps



Installing TI-RTOS

This chapter covers the steps to install TI-RTOS within Code Composer Studio or as a standalone software product.

Topic		Page
2.1	System Requirements	10
2.2	Installing Code Composer Studio	11
2.3	Installing TI-RTOS in Code Composer Studio	11
2.4	Installing TI-RTOS for Use in IAR Embedded Workbench	12
2.5	Installing TI-RTOS as a Standalone Product	12

2.1 System Requirements

The Microsoft Windows version of TI-RTOS can be installed on systems running Windows 8, Windows 7, Windows Vista, or Windows XP (SP2 or SP3).

The Linux version of TI-RTOS can be installed on systems that are running Linux RedHat v4 and higher or Ubuntu v10.04 and higher.

Separate versions of TI-RTOS are available for various Texas Instruments device families.

In order to install TI-RTOS, you must have at least 1 GB of free disk space. (If you have not yet installed Code Composer Studio, you will also need at least 4 GB of disk space for that installation.)



2.2 Installing Code Composer Studio

TI-RTOS is used in conjunction with Code Composer Studio 6.1 or higher. (TI-RTOS can also be used with the IAR Embedded Workbench IDE. See page 12 for more information.) CCS is available for Microsoft Windows and Linux.

For Windows installations, we recommend that you install CCS in the default installation directory of c:\ti. If you install in c:\Program Files (or c:\Program Files (x86) with Windows 7), you are likely to run into problems related to Windows security permissions.

Note:

Do not install CCS in a location that contains any spaces in the full path. For example, CCS should not be installed in c:\Program Files. Makefiles may not function correctly with directory paths that include spaces.

To install CCS 6.x, go to the "Download CCS" page on the Texas Instruments wiki and follow a link to download the software for your license type. For multi-user licenses, see the CCS product page.

Run the installer, and answer the prompts as appropriate.

2.3 Installing TI-RTOS in Code Composer Studio

TI-RTOS is *not* installed automatically as part of the Code Composer Studio v6.x installation. Instead, you can install it through the CCS App Center as described in this section, for use in IAR Workbench as described in Section 2.4, or as a standalone product as described in Section 2.5.

Follow these steps to install TI-RTOS in CCS:

- 1. Run CCS v6.1 or higher.
- 2. Choose View > CCS App Center in CCS.
- 3. Select the version of TI-RTOS for your device family. If you use devices from multiple families, you can select multiple TI-RTOS versions.
- 4. Click the Install Software button near the top of the App Center view.



- 5. Answer the prompts as necessary to complete the TI-RTOS installation.
- 6. Restart CCS in order for TI-RTOS and its components to be available.





2.4 Installing TI-RTOS for Use in IAR Embedded Workbench

You can install TI-RTOS for use with IAR Embedded Workbench as follows:

- Install IAR Embedded Workbench for Texas Instruments MSP430 devices. See Section 1.4 for the supported versions of IAR Workbench.
- 2. Download the Windows installer for TI-RTOS for MSP43x. For example, tirtos_setupwin32_msp43x_2_##_##_##.exe.
- 3. Run the downloaded file to install the full TI-RTOS product. You can install TI-RTOS in a standalone directory. Installing in a directory path that contains spaces, such as $C: \Program\ Files\ (x86)$, is not supported.

Note: TI-RTOS installs the core functionality of the XDCtools component if you have not already installed the necessary version as part of a CCS installation. TI-RTOS places XDCtools in a separate directory at the same level where you install TI-RTOS. For example, if the TI-RTOS installation directory is C:\ti\tirtos_msp43x_2_##_##, the XDCtools directory will be C:\ti\xdctools 3 31 ## ## core.

Follow the instructions in Section 3.1.2 to complete the installation of the TI-RTOS examples. See Section 3.2 to build the examples with the IAR compiler and to load and run the examples with IAR Embedded Workbench.

2.5 Installing TI-RTOS as a Standalone Product

If you do not use Code Composer Studio, you can install TI-RTOS as a standalone product. In addition to compiling and linking with the Texas Instruments Code Generation Tools, TI-RTOS includes support for the IAR tool chain.

- 1. Download the Windows or Linux installer for TI-RTOS for the device family you use. For example, tirtos_setupwin32_msp43x_2 ## ## ##.exe Of tirtos_setuplinux_msp43x_2 ## ## ##.bin.
- 2. Run the downloaded file to install TI-RTOS. You can install TI-RTOS in a standalone directory. Installing in a directory path that contains spaces, such as C:\Program Files (x86), is not supported.

Note: TI-RTOS installs the core functionality of the XDCtools component if you have not already installed the necessary version as part of a CCS installation. TI-RTOS places XDCtools in a separate directory at the same level where you install TI-RTOS. For example, if the TI-RTOS installation directory is located in C:\ti\tirtos_msp43x_2_##_###, the XDCtools directory will be in C:\ti\xdctools 3 31 ## ## core.

Follow the instructions in Section 3.1.2 to complete the installation of the TI-RTOS examples.



Examples for TI-RTOS

TI-RTOS comes with a number of examples that illustrate on how to use the individual components. This chapter explains how to create and use these examples.

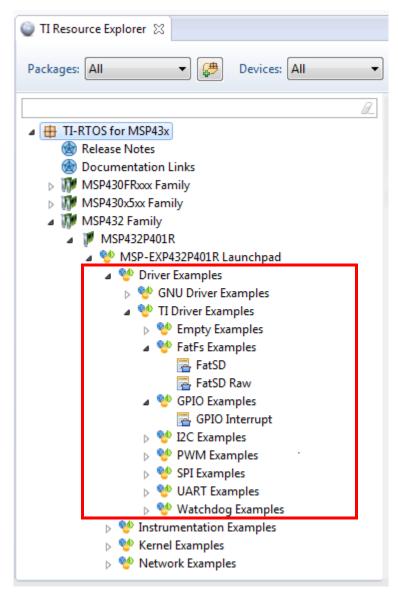
T	opic		Page
•	3.1	Creating Example Projects Using the Resource Explorer in CCS	14
;	3.2	Creating Examples with IAR Embedded Workbench	18
;	3.3	Summary of Example Peripheral Use and Target Support	18
;	3.4	MSP430 MSP-EXP430F5529LP Settings and Resources	20
;	3.5	MSP430 MSP-EXP430FR5969LP Settings and Resources	22
;	3.6	MSP430 MSP-EXP430FR6989LP Settings and Resources	24
;	3.7	MSP432 MSP-EXP432P401RLP Settings and Resources	26
;	3.8	BoosterPacks	28
	3.9	Installing USB Drivers for the USB Device Examples	32



3.1 Creating Example Projects Using the Resource Explorer in CCS

You can use the **Resource Explorer** in Code Composer Studio (CCS) to create example projects that use TI-RTOS and its components and have all the settings needed for your device. Follow these steps:

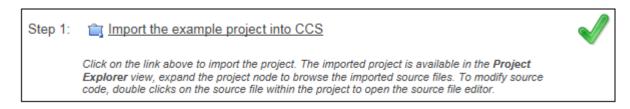
- 1. Open CCS. If you do not see the Resource Explorer, make sure you are in the CCS Edit perspective and choose View > Resource Explorer (Examples) from the menus.
- 2. Type the name or part of the name of your device in the **enter search keyword** field to hide all the examples that don't apply to your device. Or, type "Driver Examples" to find driver examples.
- 3. Expand the tree until you see the examples for your device. Any **Driver Examples** listed are TI-RTOS driver examples. Any **Instrumentation Examples** listed are UIA examples. The **Kernel Examples** are the TI-RTOS Kernel (SYS/BIOS) examples. **Network Examples** use capabilities such as WiFi, PPP, Ethernet, and HTTP/HTTPS.



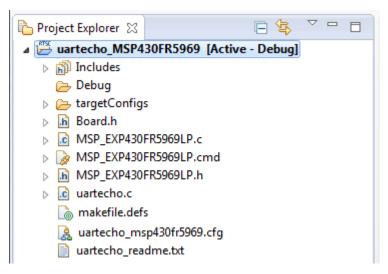
4. Select the example you want to create. A description of the selected example is shown to the right of the example list.



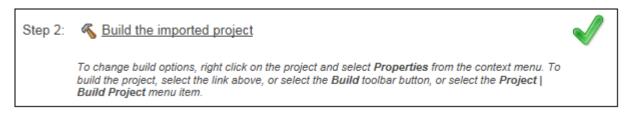
5. Click the **Step 1** link in the right pane of the Resource Explorer to **Import the example project into CCS**. This adds a new project to your Project Explorer view. Once you have completed a step for a particular example and device, a green checkmark will be shown next to that step.



6. The project created will have a name with the format < example_name>_<box>board>. You can expand the project to see the source code, configuration, and other files in the project.



7. The page shown when you select an example in the Resource Explorer provides additional links to perform common actions with that example. Use the **Step 2** link when you are ready to build the project. If you want to change any build options, right click on the project and select **Properties** from the context menu. For example, you can change compiler, linker, and RTSC (XDCtools) options.

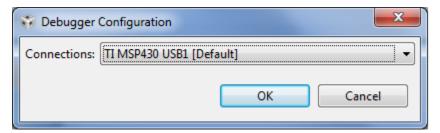




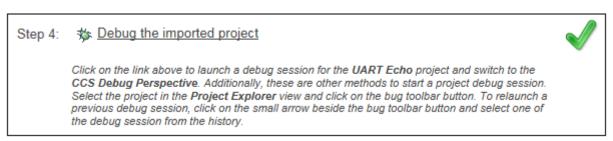
8. Use the **Step 3** link to change the connection used to communicate with the board. The current setting is shown in the Resource Explorer page for the selected example. (If you want to use a simulator instead of a hardware connection, double-click the *.ccxml file in the targetConfigs folder of the project to open the Target Configuration File editor. Change the **Connection** as needed, and click **Save**.)



9. You will see the Debugger Configuration dialog. Choose an emulator. For MSP430 devices, choose the **TI MSP430 USB1**, and for MSP432 devices, choose the **TI XDS 110 USB Emulator**.



10. Use the **Step 4** link to launch a debug session for the project and switch to the **CCS Debug** Perspective.



3.1.1 Creating an Empty TI-RTOS Project

TI-RTOS provides blank projects you can use as a starting point for creating your own projects that utilize TI-RTOS. Both "Empty" and "Empty (Minimal)" versions are provided. The "Empty" version enables more kernel features and debug capabilities at the cost of large footprint. The "Empty (Minimal)" version disables various kernel features and debug capabilities to minimize the footprint. See the "Memory Usage with TI-RTOS" chapter in the TI-RTOS User's Guide (SPRUHD4) for details about techniques used to minimize the footprint.

Empty TI-RTOS driver projects can be created with the Resource Explorer (see Section 3.1).

After you create the example, the files in the empty project example include:

- Key C files: empty.c, <board>.c/.h
- Key configuration files: empty.cfg
- Linker command file: <board>.cmd

Add to the example as needed to implement your application.



3.1.2 Creating Examples to Build via a Command Line

TI-RTOS has a command-line utility called <code>examplesgen</code> that generates example projects along with the makefiles needed to build the examples for all supported tool chains—TI and IAR. The files are created in a location you specify on the command line. The <code>tirtos.mak</code> file in the top level directory of your TI-RTOS installation can be used to run examplesgen.

Note:

If you installed TI-RTOS using the standalone installer (Section 2.5 or Section 2.4), this step is not necessary because pre-generated examples are included as part of the installation for all supported boards and tool chains. The provided examples are located in the <code>TIRTOS_INSTALL_DIR\tirtos_msp43x_2_##_##_##_examples</code> directory. Pre-generated examples are not provided if you installed TI-RTOS through the CCS App Center.

You only need to perform these steps once:

- 1. If you installed TI-RTOS in a location other than the default location of C:\ti, edit the tirtos.mak file in the TI-RTOS installation directory. Modify the following variables as needed to make them point to the correct locations.
 - DEFAULT INSTALLATION DIR: Full path to the location where TI tools are installed.
 - IAR_COMPILER_INSTALLATION_DIR: Full path to the IAR code generation tools installation.
 - GCC_INSTALLATION_DIR: Full path to the GCC code generation tools installation. (GCC is not supported for this device family.)
 - TIRTOS_INSTALLATION_DIR: Full path to the TI-RTOS installation.
 - XDCTOOLS INSTALLATION DIR: Full path to the XDCtools installation.
- 2. If you plan to use TI-RTOS with IAR, set the IAR_BUILD variable in the tirtos.mak file to true.

```
IAR_BUILD ?= true
```

3. Open a command line window, and use the following commands to run the examplesgen utility. (If you installed TI-RTOS in a protected directory, you should run the command window as the administrator.)

```
> cd <tirtos_install>
> ..\xdctools_3_31_##_##_core\gmake -f tirtos.mak examplesgen DEST="YOURPATH"
```

For the destination path, use a UNIX-style path. That is, use forward slashes (/) instead of backslashes (\). For example, DEST="C:/myfiles".

The output from this command is a tirtos_msp43x_2_##_##_##_examples directory tree containing folders for the supported boards. Each board folder contains folders for all the examples available for that board.

Examples for TI and IAR are generated for boards supported by TI-RTOS. Each board directory contains a makedefs file that can be modified to specify other installation paths or compiler/linker options and a makefile that can be used to build all the examples for that board. Each example directory has its own makefile that can be used to build that example specifically.



3.2 Creating Examples with IAR Embedded Workbench

For information about using TI-RTOS examples with IAR Embedded Workbench, see the wiki page on Creating TI-RTOS Applications in IAR Embedded Workbench on the Texas Instruments wiki.

3.3 Summary of Example Peripheral Use and Target Support

The boards, components, and hardware used by the TI-RTOS examples are shown in the following table. Details about these examples are provided in the readme files in the example projects.

There are several example categories. The Empty and Empty (Minimal) projects are configured to make TI-RTOS available but do not contain specific code that uses TI-RTOS. The remaining examples show how to use a specific peripheral.

Table 3-1. TI-RTOS for MSP43x Examples: Boards, Components, and Drivers

Example	MSP-EXP430F5529LP	MSP-EXP430FR5969LP	MSP-EXP430FR6989LP	MSP-EXP432P401RLP	TI-RTOS Kernel (SYS/BIOS)	SD Card / SDSPI	USB HID	USB CDC	1 ² C	GPIO	PWM	SPI	Watchdog Timer	WiFi	UART	UIA
Empty and Empty (Minimal) TI-RTOS Projects	Х	Х	Х	Х	Х					Х						Χ
Graphics Library Demo		Χ	Х		Х	X				Х					Χ	
TCP Echo for CC3100 ¹	Х	Х	Х	Х	Х					Х		Х		Х		
UDP Echo for CC3100 ¹	Х	Х	Х	Х	Х					Х		Х		Х		
SPI Loopback				Х	Х					Х		Χ				
FatSD: FatFs File Copy	x ²	x ²	x ²	Х	Х	Х				Х						
FatSD Raw: FatFs File Copy using FatFs APIs	x ²	x ²	x ²	Х	Х	Х				Х						
GPIO Interrupt	Χ	Х	Х	Х	Х					Х						
I ² C RF430CL330 Load ³	Х	Х	Х	Х	Х				Х	Х						
I ² C TMP006 ⁴	Х	Х	Х	Х	Х				Х	Х						
I ² C TPL0401EVM ⁵	Х	Х	Х	Х	Х				Х	Х						
PWM LED				Х	Х					Х	Х					
UART Console	Х				Х			Х		Х					Х	Х
UART Echo	Χ	Χ	Χ	Х	Х					Χ					Χ	Χ
UART Logging	Х	Х	Х	Х	Х					Х					Х	Χ
USB Keyboard Device	Χ				Х		Χ			Χ						
USB Mouse Device	Χ				Х		Х			Х						



Example	MSP-EXP430F5529LP	MSP-EXP430FR5969LP	MSP-EXP430FR6989LP	MSP-EXP432P401RLP	TI-RTOS Kernel (SYS/BIOS)	SD Card / SDSPI	USB HID	USB CDC	l²C	GPIO	PWM	IdS	Watchdog Timer	WiFi	UART	UIA
USB Serial Device	Χ				Χ			Χ		Χ						
USB CDC Mouse Device	Χ				Χ		Χ	Χ		Χ						
Watchdog	Χ	Х	Х	Х	Х					Χ			Χ			

¹This example requires either a CC3100 EM board or CC3100 BoosterPack. See the "Jumper Settings: section in Section 3.4 through Section 3.5 for details.

3.3.1 Readme Files and Common Driver Example Features

There is a separate <example_name>_readme file for each of the examples. These files are added to your CCS project when you use the Resource Explorer to create a project. You can open the <example_name>_readme file within CCS. The <example_name>_readme files contain the following types of information:

- Actions performed by functions in the example.
- Hardware-specific descriptions of buttons, LEDs, etc...
- Which external components are (or may be) needed to run with particular examples.

The Driver Examples share the following features:

- Most TI-RTOS driver examples use the SysMin System Support module. See the readme files in the individual example projects for details.
- The empty, demo, and most UART examples use the ti.uia.sysbios.LoggingSetup module with stop
 mode data collection. The UART Console example uses run-time data collection during Idle thread
 processing. For more details on data collection, see Chapter 2 of the TI-RTOS User's Guide
 (SPRUHD4).
- Driver Examples for a particular target all have the same <board>.c and <board>.h files. These files perform board-specific configuration of the drivers provided by TI-RTOS. For more details, see Chapter 4 of the TI-RTOS User's Guide (SPRUHD4).

²This example requires an SD Card reader BoosterPack.

³This example requires a RF430CL330 NFC transponder module and the Wireless Connectivity Adaptor. See Section 3.8.5.

⁴This example requires a TMP006 BoosterPack. See Section 3.8.2.

⁵This example requires a TPL0401EVM BoosterPack. See Section 3.8.6.

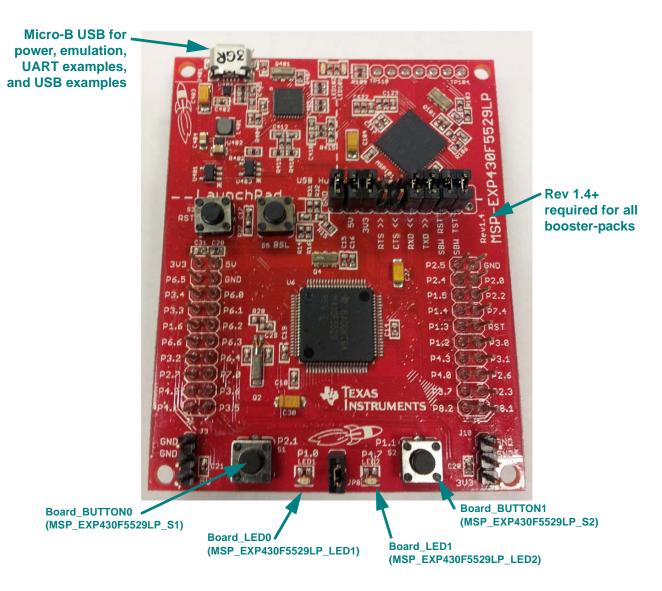


The sections that follow list settings required to run the TI-RTOS examples on the supported boards. They also list the hardware resources that TI-RTOS and its dependent components use by default. Some of these resources offer flexible options, whereas others are fixed in the current design or implementation.

3.4 MSP430 MSP-EXP430F5529LP Settings and Resources

The MSP-EXP430F5529LP board contains an MSP430 MSP430F5529 device.

MSP-EXP430F5529LP connections:



Jumper settings:

Set RXD << and TXD >> to provide UART communications via the onboard USB debugger.

If you use a BoosterPack with this board (Section 3.8), the BoosterPack fits over J1 and J5.



SPI Loopback example pin connections:

The SPI loopback example is not supported on MSP430F5529.

Resources Used:

The following list shows which MSP_EXP430F5529LP resources are used by TI-RTOS applications on that platform. TI-RTOS examples control which peripherals (and which ports) are used.

• TI-RTOS Kernel (SYS/BIOS). Uses the first general-purpose timer available and that timer's associated interrupts. Generally, this will be TimerO_A0. The TI-RTOS Kernel manages the interrupt controller statically without an interrupt dispatcher.

TI-RTOS.

- SD Card. Uses FatFs and the SDSPI driver on USCI_B0 without interrupts to read and write to files.
- I²C. The I²C driver is configured on USCI B0 to support various BoosterPacks.
- **GPIOs.** The GPIO driver is used on 2 onboard LEDs: LED1 (P1.0) and LED2 (P4.7) as output pins and 2 switches S1 P2.1 and S2 P1.1 as inputs.
- Serial. The UART driver uses USCI_A1, which is attached to the onboard EZ-FET MSP430 USB chip to facilitate serial communications.
- SPI. The SPI driver is configured to use USCI_B1 and USCI_B0 for SPI communications. Each
 instance requires 2 DMA channels; therefore, only one SPI instance can be used at a time.
- USB. The USB reference examples use the USB library and the USB controller with its associated interrupts.
- **Watchdog.** The Watchdog driver example uses Watchdog Timer A.
- WiFi. The WiFi examples for use with the CC3100 use GPIO pins P2.0, P2.2, and P6.5 as well as the GPIO interrupt for port 2. In addition, it uses P3.0 and P3.2 for SPI communications.

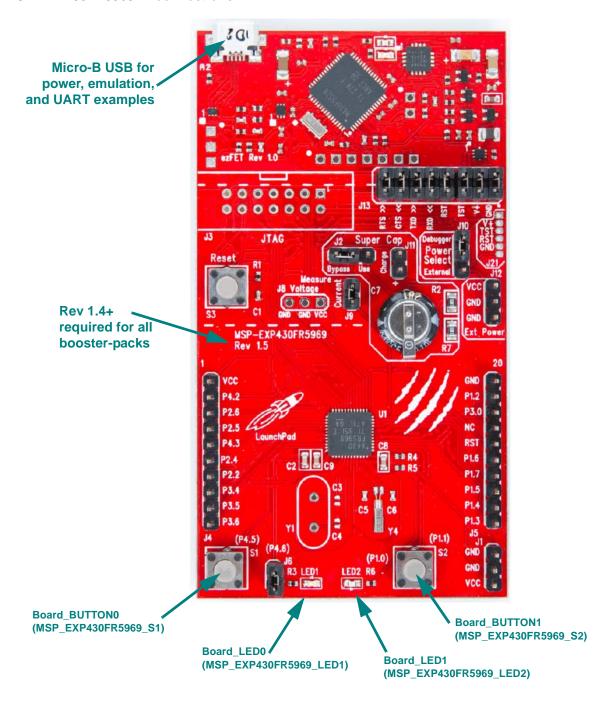
For MSP430, USB drivers are stored in the USB_config folder in TI-RTOS USB examples in CCS. The USB reference modules for MSP430 use a set of descriptor files that were generated with the MSP430 USB Descriptor Tool.



3.5 MSP430 MSP-EXP430FR5969LP Settings and Resources

The MSP-EXP430FR5969LP board contains an MSP430FR5969 device. This device has an FRAM memory unit.

MSP-EXP430FR5969LP connections:





Jumper settings:

Set RXD << and TXD >> to provide UART communications via the onboard USB debugger.

If you use a BoosterPack with this board (Section 3.8), the BoosterPack fits over J4 and J5.

SPI Loopback example pin connections:

The SPI loopback example is not supported on the MSP_EXP430FR5969LP.

Resources Used:

The following list shows which MSP_EXP430FR5969LP resources are used by TI-RTOS applications on that platform. TI-RTOS examples control which peripherals (and which ports) are used.

• TI-RTOS Kernel (SYS/BIOS). Uses the first general-purpose timer available and that timer's associated interrupts. Generally, this will be Timer0_A3. The TI-RTOS Kernel manages the interrupt controller statically without an interrupt dispatcher.

TI-RTOS.

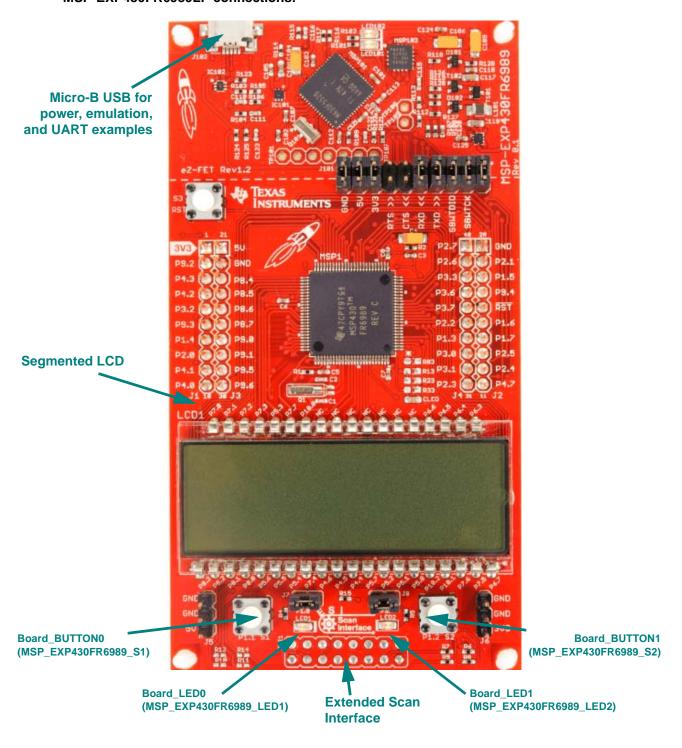
- SD Card. Uses FatFs and the SDSPI driver on EUSCI_B0 without interrupts to read and write to files
- I²C. The I²C driver is configured on EUSCI_B0 to support various BoosterPacks.
- GPIOs. The GPIO driver is used on 2 onboard LEDs: LED1 (P4.6) and LED2 (P1.0) as output pins and 2 switches S1 P4.5 and S2 P1.1 as inputs.
- Serial. The UART driver uses EUSCI_A0, which is attached to the onboard EZ-FET MSP430 USB chip to facilitate serial communications.
- **SPI.** The SPI driver is configured to use EUSCI_B0 for SPI communications. This instance requires 2 DMA channels for the SPI transfer, one for transfer and one for receive.
- Watchdog. The Watchdog driver example uses Watchdog Timer A.
- **WiFi.** The WiFi examples for use with the CC3100 use GPIO pins P1.2, P3.0, and P4.2 as well as the GPIO interrupt for port 1. In addition, it uses the EUSCI B0 pins for SPI communications.



3.6 MSP430 MSP-EXP430FR6989LP Settings and Resources

The MSP-EXP430FR6989LP board contains an MSP430FR6989 device. This device has an FRAM memory unit.

MSP-EXP430FR6989LP connections:





Jumper settings:

Set RXD << and TXD >> to provide UART communications via the onboard USB debugger.

If you use a BoosterPack with this board (Section 3.8), the BoosterPack fits over J1 through J4.

SPI Loopback example pin connections:

The SPI loopback example is not supported on the MSP EXP430FR6989LP.

Resources Used:

The following list shows which MSP_EXP430FR6989LP resources are used by TI-RTOS applications on that platform. TI-RTOS examples control which peripherals (and which ports) are used.

• TI-RTOS Kernel (SYS/BIOS). Uses the first general-purpose timer available and that timer's associated interrupts. Generally, this will be Timer0_A0. The TI-RTOS Kernel manages the interrupt controller statically without an interrupt dispatcher.

TI-RTOS.

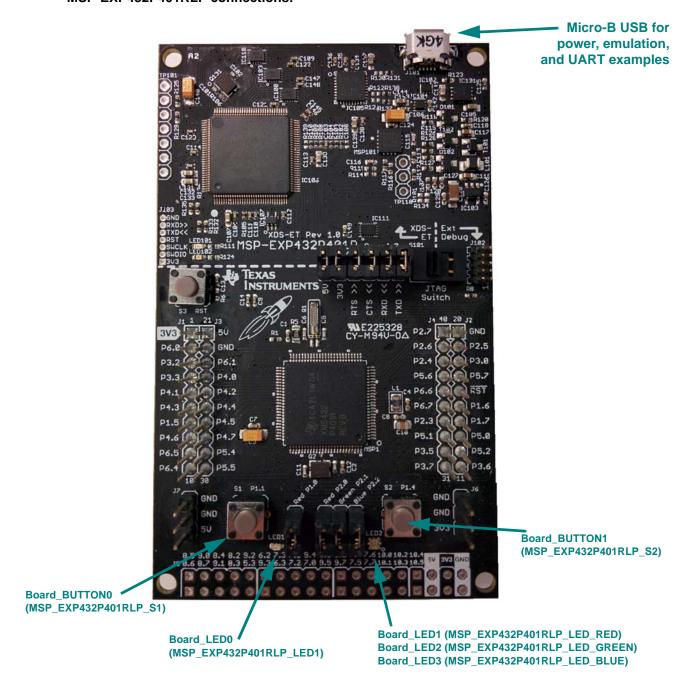
- SD Card. Uses FatFs and the SDSPI driver on EUSCI_B0 without interrupts to read and write to files
- I²C. The I²C driver is configured on EUSCI_B0 to support various BoosterPacks.
- GPIOs. The GPIO driver is used on 2 onboard LEDs: LED1 (P1.0) and LED2 (P9.7) as output pins and 2 switches S1 P1.1 and S2 P1.2 as inputs.
- Serial. The UART driver uses EUSCI_A1, which is attached to the onboard EZ-FET MSP430 USB chip to facilitate serial communications.
- **SPI.** The SPI driver is configured to use EUSCI_B0 for SPI communications. This instance requires 2 DMA channels for the SPI transfer, one for transfer and one for receive.
- Watchdog. The Watchdog driver example uses Watchdog Timer A.
- **WiFi.** The WiFi examples for use with the CC3100 use GPIO pins P1.5, P3.2, and P2.1 as well as the GPIO interrupt for port 2. In addition, it uses the EUSCI B0 pins for SPI communications.



3.7 MSP432 MSP-EXP432P401RLP Settings and Resources

The MSP-EXP432P401RLP board contains an MSP432P401R device.

MSP-EXP432P401RLP connections:





Jumper settings:

- Set RXD << and TXD >> to provide UART communications via the onboard USB debugger.
- If you use a BoosterPack with this board (Section 3.8), the BoosterPack fits over J1 and J5.

SPI Loopback example pin connections:

When wiring for SPI loopback, pins on a single board are wired to other pins on the same board.

Master Pin (Function)	Slave Pin (Function)
P1.5 (EUSCI_B0_CLK) —	- P3.5 (EUSCI_B1_CLK)
P1.6 (EUSCI_B0_SIMO) ——	P3.6 (EUSCI_B1_SIMO)
P1.7 (EUSCI_B0_SOMI) ——	- P3.7 (EUSCI_B1_SOMI)

Resources Used:

The following list shows which MSP_EXP432P401RLP resources are used by TI-RTOS applications. TI-RTOS examples control which peripherals (and which ports) are used.

• TI-RTOS Kernel (SYS/BIOS). Uses the first general-purpose timer available and that timer's associated interrupts. Generally, this will be Timer0_A0. The TI-RTOS Kernel manages the interrupt controller statically without an interrupt dispatcher.

TI-RTOS.

- GPIO. The GPIO driver is used on 2 onboard LEDs: LED1 (P1.0) and LED2 (P2.0) as output pins and 2 switches S1 (P1.1) and S2 (P1.4) as inputs.
- I²C. The I²C driver is configured on EUSCI_B0 to support various BoosterPacks.
- PWM. The PWM driver uses the onboard RGB LED (G:P2.1 and B:P2.2). These pins are
 configured for the PWM driver. While these pins can also be used by the GPIO driver, your
 application's board file must ensure that the pins are not simultaneously used by the GPIO and
 PWM drivers.
- SD Card. Uses FatFs and the SDSPI driver on EUSCI_B0 without interrupts to read and write to files.
- **SPI.** The SPI driver is configured to use EUSCI B0 and EUSCI B1 for SPI communications.
- Serial. The UART driver uses EUSCI_A0, which is attached to the onboard emulator to facilitate serial communications.
- Watchdog. The Watchdog driver example uses the Watchdog Timer A peripheral.
- WiFi. The WiFi examples for the CC3100 use the following GPIO pins: P4.1 (enable), P3.0 (chip select), and P2.5 as the CC3100 interrupt pin (interrupt is created for the port). The driver also uses an SPI driver instance (EUSCI_B0) and associated DMA channels.



BoosterPacks www ti com

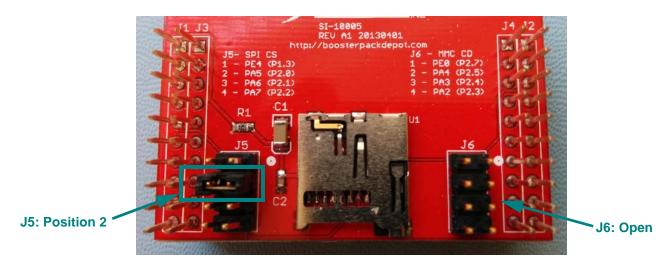
3.8 **BoosterPacks**

Several BoosterPack boards are used with the TI-RTOS examples. This section described those boards and provides any special notes about installing the board.

3.8.1 SD Card BoosterPack

A microSD BoosterPack or SD Card BoosterPack should be used with examples that require an SD Card reader on target boards that do not include an SD Card reader. This board may be used with the MSP-EXP430F5529LP, MSP-EXP430FR5969LP, MSP-EXP430FR6989LP, and MSP-EXP432P401RLP.

The figure below shows the jumper needed on the microSD BoosterPack to make it pin-compatible with the SD Card BoosterPack.



3.8.2 TMP006 BoosterPack

The I²C TMP006 example uses the TMP006 BoosterPack to connect the TMP006EVM circuit board to the LaunchPad. The TMP006 BoosterPack kit, which includes a TMP006EVM circuit board, is available online. This board may be used with the MSP-EXP430F5529LP, MSP-EXP430FR5969LP, MSP-EXP430FR6989LP, and MSP-EXP432P401RLP.

3.8.3 Sharp Memory LCD BoosterPack

The LCD BoosterPack has a 96 x 96 pixel display that should be used with the graphics library demo example on the MSP-EXP430FR5969LP. This board may also be used with the MSP-EXP430F5529LP and MSP-EXP432P401RLP for applications you create. It is available on the TI eStore.

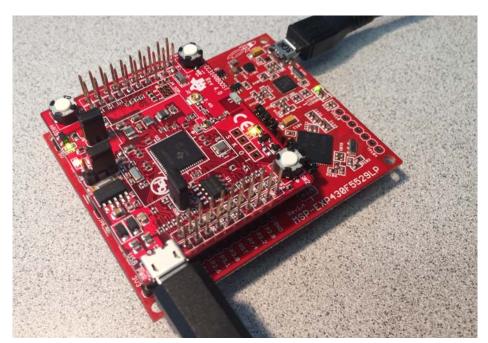
www.ti.com BoosterPacks

3.8.4 CC3100 BoosterPacks

The TI CC3100 is a self-contained wireless network processor that simplifies the implementation of Internet connectivity. TI's SimpleLink Wi-Fi solution minimizes the software requirements of the host microcontroller (MCU) and is thus the ideal solution for embedded applications using any low-cost and low-power MCU.

The CC3100 BoosterPack is available to be used with the following boards (pictures are provided to show the correct way to mount the booster pack onto each board):

MSP-EXP430F5529LP



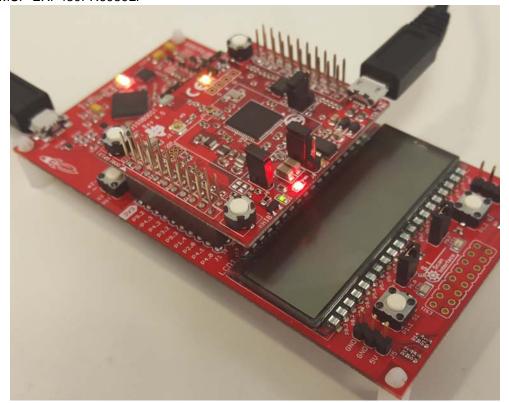
MSP-EXP430FR5969LP



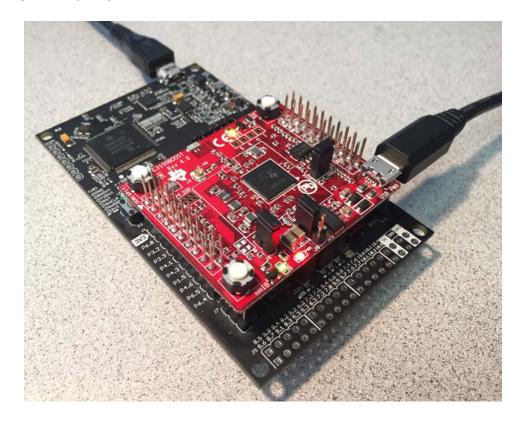


BoosterPacks www.ti.com

MSP-EXP430FR6989LP



MSP-EXP432P401R





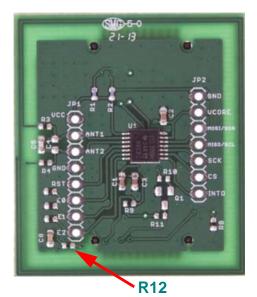
www.ti.com BoosterPacks

3.8.5 RF430CL330 NFC Transponder Module

The I²C RF430CL330 Load example uses the I²C driver to communicate with a RF430CL330 NFC transponder module. This board may be used with the MSP-EXP430F5529LP, MSP-EXP430FR5969LP, MSP-EXP430FR6989LP, and MSP-EXP432P401RLP.

A Wireless Connectivity Adaptor board is also required in order to use the NFC Transponder Module with a LaunchPad board.

Hardware Modification Note: In order to use the RF430CL330 module with a Wireless Connectivity Adapter board, you need to remove R12 from the RF430CL330H board.

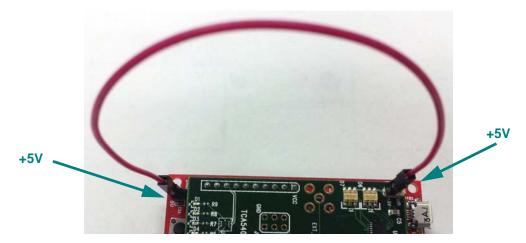




3.8.6 TPL0401EVM Board

The I²C TPL0401EVM example requires an attached TPL0401EVM board. This board may be used with the MSP-EXP430F5529LP, MSP-EXP430FR5969LP, MSP-EXP430FR6989LP, and MSP-EXP432P401RLP.

A wire like the one shown in the following figure should be added to provide power to the TPL0401EVM board. This figure shows the wire on an MSP-EXP430F5529LP board. Similar wiring is required to connect the TPL0401EVM board to other boards supported by this example. A 5V power supply is recommended to provide sufficient LED brightness.



3.9 Installing USB Drivers for the USB Device Examples

The USB examples build upon the examples provided with MSPWare. Because the examples mimic the same functionality, you can use the same drivers delivered with standalone installations of MSPWare.

USB drivers for MSP430 are located in each example's USB config folder.

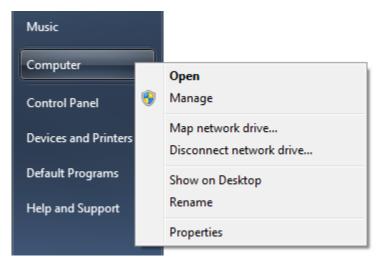
The Windows menus and dialogs you see may be slightly different from those shown here, depending on your version of Windows.

To install the USB driver, follow these steps:

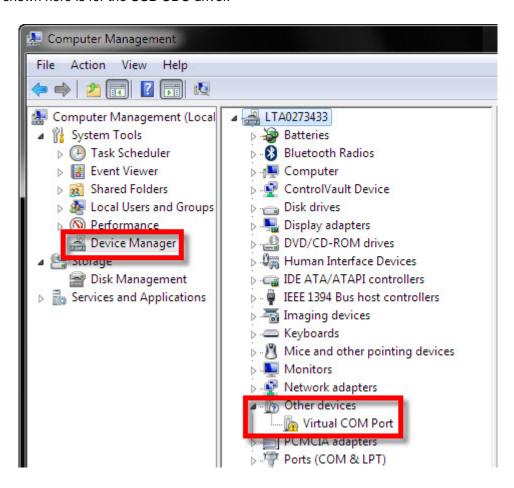
- Load and run a USB device reference example—USB Keyboard Device, USB Mouse Device, or USB CDC Mouse Device.
- 2. While the example is running, connect the device to the Windows PC via a USB cable. At this point, Windows will detect the device and attempt to enumerate it.



3. Open the Windows Device Manager by right-clicking on the **My Computer** desktop icon or **Computer** in the Start Menu and selecting **Manage**.



- 4. If you are prompted by a security warning in Windows 7, click Yes.
- 5. Select the **Device Manager** category in the left pane.
- 6. In the center pane, select the unknown driver that you are trying to install. For example, the device shown here is for the USB CDC driver.





- 7. Right-click on the device, and select **Update Driver Software**.
- 8. Select **Browse my computer for driver software** and browse to the location of the Windows USB drivers. Make sure the box to **Include subfolders** is checked.
- 9. Click Next to run the installation wizard. If you see a Window Security prompt, click Install.
- 10. After the driver is installed, you can determine the COM port number for the CDC (Virtual COM Port) device.



Configuring TI-RTOS

This chapter describes how to configure TI-RTOS and its components for use by your application.

Topic		Page
4.1	Starting the Configuration Tool	36
4.2	Configuring TI-RTOS Drivers	37
4.3	Configuring Components of TI-RTOS	38



4.1 Starting the Configuration Tool

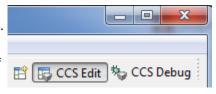
Note:

The graphical configuration tool is not available within IAR Embedded Workbench. If you are using IAR, edit the project's *.cfg file within IAR as a text-based source file. See the Texas Instruments Wiki for more about using IAR with TI-RTOS.

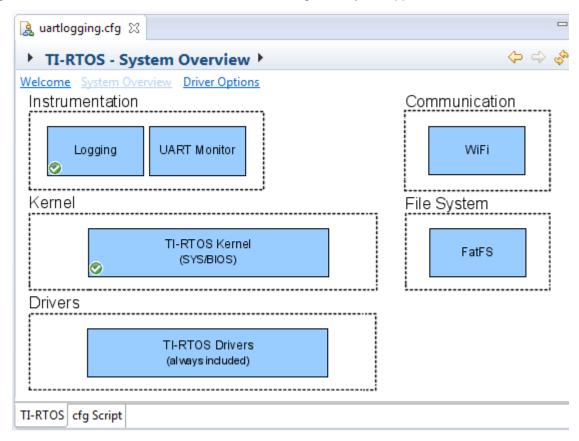
This section shows how to open the Graphical Configuration Tool (XGCONF) to view the System Overview. For details on using XGCONF, see Chapter 2 of the SYS/BIOS User's Guide (SPRUEX3).

To use CCS to open the graphical tool for editing configuration files (XGCONF), follow these steps:

- 1. Make sure you are in the **CCS Edit** perspective of CCS. If you are not in that perspective, click the CCS Edit icon to switch back.
- Double-click on the *.cfg configuration file for a TI-RTOS
 example project in the **Project Explorer** tree. (See Section 3.1 if
 you need to create an example project.) While XGCONF is
 opening, the CCS status bar shows that the configuration is
 being processed and validated.



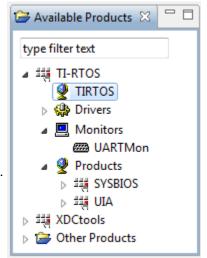
- When XGCONF opens, you see the **Welcome** sheet for TI-RTOS if you are using a Driver example. (If this is the configuration file for a Kernel example or an Instrumentation example, the Welcome sheet for SYS/BIOS opens first, instead.) The Welcome sheet provides links to documentation resources.
- 4. Click the System Overview link to see a diagram of the components available through TI-RTOS. (SYS/BIOS modules are shown if you are using a Kernel example or an Instrumentation example.) A green check mark indicates which modules are being used by the application.





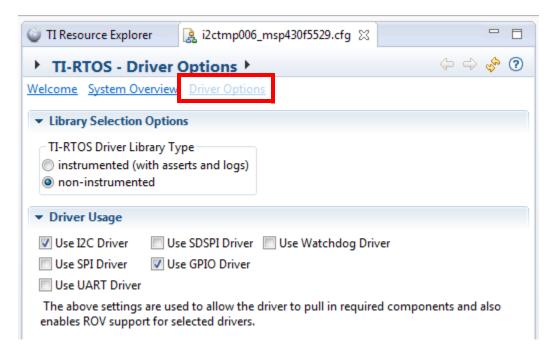
- 5. You also see a list of Available Products in a pane on the left of the CCS window. This list allows you to select the TIRTOS module and any configurable modules in the products TI-RTOS provides.
- 6. Click a blue box in the System Overview to go to the configuration page for a module.

Note: If the configuration is shown in a text editor instead of XGCONF, close the text editor window. Then, right-click on the *.cfg file and choose **Open With > XGCONF**. If you are comfortable editing configuration scripts with a text editor, you can do that. However, you should not have the file open in both types of editor at the same time.



4.2 Configuring TI-RTOS Drivers

In the System Overview display for the TIRTOS module, select the **Driver Options** link.



You can choose to use either the instrumented or non-instrumented driver libraries when linking with TI-RTOS. The instrumented libraries process Log events while the non-instrumented libraries do not. See the section on "Using Instrumented or Non-Instrumented Libraries" in the *TI-RTOS User's Guide* (SPRUHD4) for more information. This setting affects all the TI-RTOS drivers listed in Section 1.6 together.

Check the boxes for any drivers your application will use. For most drivers, this is the only configuration necessary. Enabling the drivers here also enables the instrumentation provided to the RTOS Object View (ROV) plugins.

The WiFi driver is individually configurable. See the section on Static Configuration of the WiFi Driver in the *TI-RTOS User's Guide* (SPRUHD4) for details.



4.2.1 Configuring System Support

The SysCallback module lets you configure the functions that handle System output—for example, System_printf() and System_abort(). This module handles transmissions to System output only; it does not handle responses received. See the chapter on "TI-RTOS Utilities" in the *TI-RTOS User's Guide* (SPRUHD4) for more about the SysCallback module.

Other SystemSupport implementations are provided with XDCtools.

- **SysMin** stores System_printf() strings in an internal buffer in RAM. SysMin requires RAM, so it not ideal for devices with minimal RAM.
- SysStd writes System_printf() strings to STDOUT (the CCS Console window). By default, SysStd allows System_printf() to be called from Tasks only (not Swis or hardware interrupts); it can be modified to allow calls from Swis and Hwis, but this impacts real-time performance.

4.3 Configuring Components of TI-RTOS

For information about configuring individual sub-components of TI-RTOS, see the documentation for that component. Chapter 2 of the SYS/BIOS User's Guide (SPRUEX3) provides details about XGCONF. Within XGCONF, you can see the full file path to the version of the component being used by hovering your mouse cursor over a component in the "Other Products" list in the **Available Products** area.

The Resource Explorer (in CCS) lets you access documentation for all the components of TI-RTOS. Choose the **Documentation Links** item for your version of TI-RTOS. Use the configuration reference documentation, and click the "Configuration settings" link at the top of each page to skip to the configuration information.



Index

Α

App Center 11 Available Products list 37

В

BoosterPacks 28 RF430CL330 31 SD Card 28 TMP006 28

C

CCS
creating a project 6, 14
installation 11
other documentation 8
CCS App Center 4, 11
CDC device 8
example 18
components 5
Concerto
other documentation 9
configuration 35
graphical editor 36

D

disk space 10 documentation 8

E

Empty Project example 16 empty.c file 16 examples 13 overview 18

F

FatFs API, documentation 9 forum 8

G

GPIO driver 7 examples 18 resources used 21, 23, 25, 27

Н

HID device 8 example 18

ı

I2C driver 7
example 18
installation
CCS 11
directory 11
instrumentation 5
instrumented libraries 37

K

kernel examples 6

L

LaunchPad
MSP-EXP430F5529LP 20
LEDs
managed by GPIO driver 7
LoggingSetup module 19

M

MSP430F5529 7, 20 MSP430FR5969 7, 22, 26 MSP430FR6989 7, 24 MSP430Ware 5 MSP432P401R 7 MSP-EXP430F5529LP 7, 20 MSP-EXP430FR5969LP 7, 22, 26 MSP-EXP430FR6989LP 7, 24 MSP-EXP432P401RLP 7



www.ti.com

N

non-instrumented libraries 37

P

products directory 5

R

readme.txt file 19 Resource Explorer 6, 14 RF430CL330 module 31

S

SD Card BoosterPack 28
SD driver, example 18
SDSPI driver 8
resources used 21, 23, 25, 27
SPI driver 7
examples 18
resources used 21, 23, 25, 27
SYS/BIOS 5
examples 18
other documentation 8
SysCallback module, configuration 38
SysMin module 19
System Overview configuration 36
system requirements 10

Т

target configuration file 16

TMP006 BoosterPack 28 TPL0401EVM board 32

U

UART driver 8
examples 18
resources used 21, 23, 25, 27
UIA 5
example 18
other documentation 9
USB Descriptor Tool 21
USB driver
resources used 21
USB drivers 8
USB_config folder 21

W

Watchdog driver 8
examples 18
resources used 21, 23, 25, 27
WiFi driver 8
examples 18
resources used 21, 23, 25, 27
wiki 8

X

XDCtools 5 other documentation 8 XGCONF configuring other components 38 starting 36

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have not been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products	Applications
1 100000	Applications

Wireless Connectivity

Audio www.ti.com/audio Automotive and Transportation www.ti.com/automotive **Amplifiers** amplifier.ti.com Communications and Telecom www.ti.com/communications **Data Converters** dataconverter.ti.com Computers and Peripherals www.ti.com/computers **DLP® Products** www.dlp.com Consumer Electronics www.ti.com/consumer-apps DSP dsp.ti.com **Energy and Lighting** www.ti.com/energy Clocks and Timers www.ti.com/clocks Industrial www.ti.com/industrial Interface interface.ti.com Medical www.ti.com/medical Logic logic.ti.com Security www.ti.com/security

Power Mgmt power.ti.com Space, Avionics and Defense www.ti.com/space-avionics-defense

Microcontrollers microcontroller.ti.com Video & Imaging www.ti.com/video

RFID www.ti-rfid.com

www.ti.com/wirelessconnectivity

OMAP Mobile Processors www.ti.com/omap **TI E2E Community** e2e.ti.com