TI Designs
ZNP Host Framework Design Guide

TI Designs

TI Designs provide the foundation that you need including methodology, testing, and design files to quickly evaluate and customize the system. TI Designs help *you* accelerate your time to market.

Design Resources

TIDC-ZNP-HOST-SW3	Tool Folder Containing Design Files
CC2538EM	Product Folder

TI E2E™ Community

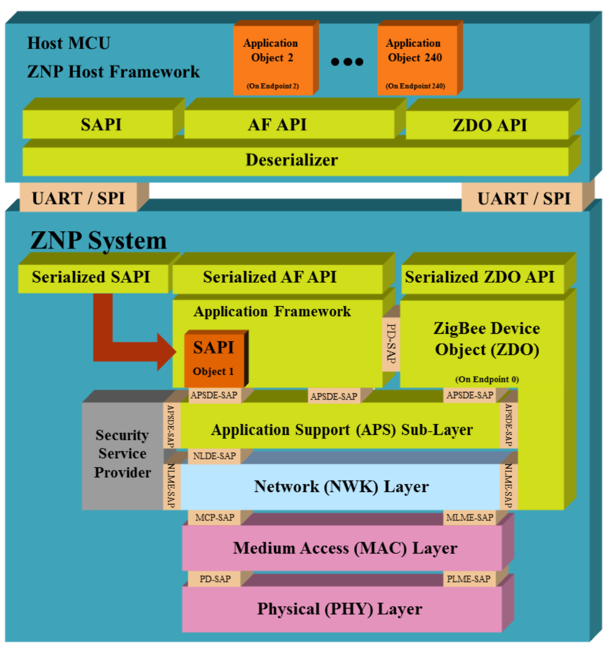
ASK Our E2E Experts
WEBENCH® Calculator Tools

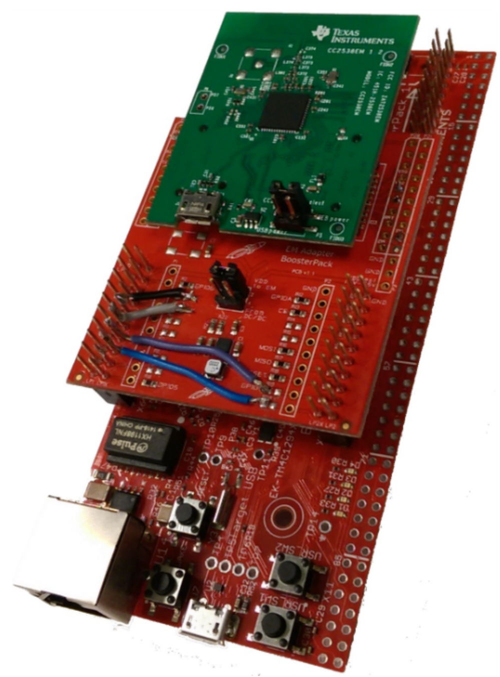
Design Features

- Portability Across Hardware Platforms
- High Level Operating System (OS), Real-Time OS Port, and OS Portable Operating System Interface (POSIX) Standard Compliance
- Intuitive Sample Codes and Examples From Command Line

Featured Applications

- Building Automation
- Factory Automation
- Sensor Monitoring and Networking





An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 Key System Specifications

Table 1. System Specifications

PARAMETER	SPECIFICATION
Platforms	POSIX, TI-RTOS
ZigBee® network processor (ZNP) hardware	CC253x
Host hardware	PC, Tiva LaunchPad™

2 System Description

The ZNP host framework is a cross-platform framework designed using the C programming language to run in a companion host microcontroller (MCU) or microprocessor unit (MPU). The framework is used in combination with the embedded Z-Stack™ ZNP products. The main motivation behind this multi-platform host framework is to provide a system that is both simple to use and simple to integrate within the most common customer platforms, enabling both quick evaluation and fast prototyping. A library of examples accompaniesy the framework, and these examples demonstrate its ease of use and familiarize users with the usage of the ZNP application program interface (API). There are four examples included in the ZNP host framework:

(a) **Command line trainer**: Command line application that provides all ZNP commands, such that the user can send any ZNP command and parameters to learn the usage of the ZNP interface.

(b) **Data send and receive**: Provides a command line interface, which allows ZNP devices to send and receive text messages back and forth.

(c) **Network topology**: Provides a description of the network topology to which the ZNP device belongs.

(d) **Service discovery**: Displays a description of the endpoints from the devices that join the network.

3 Block Diagram

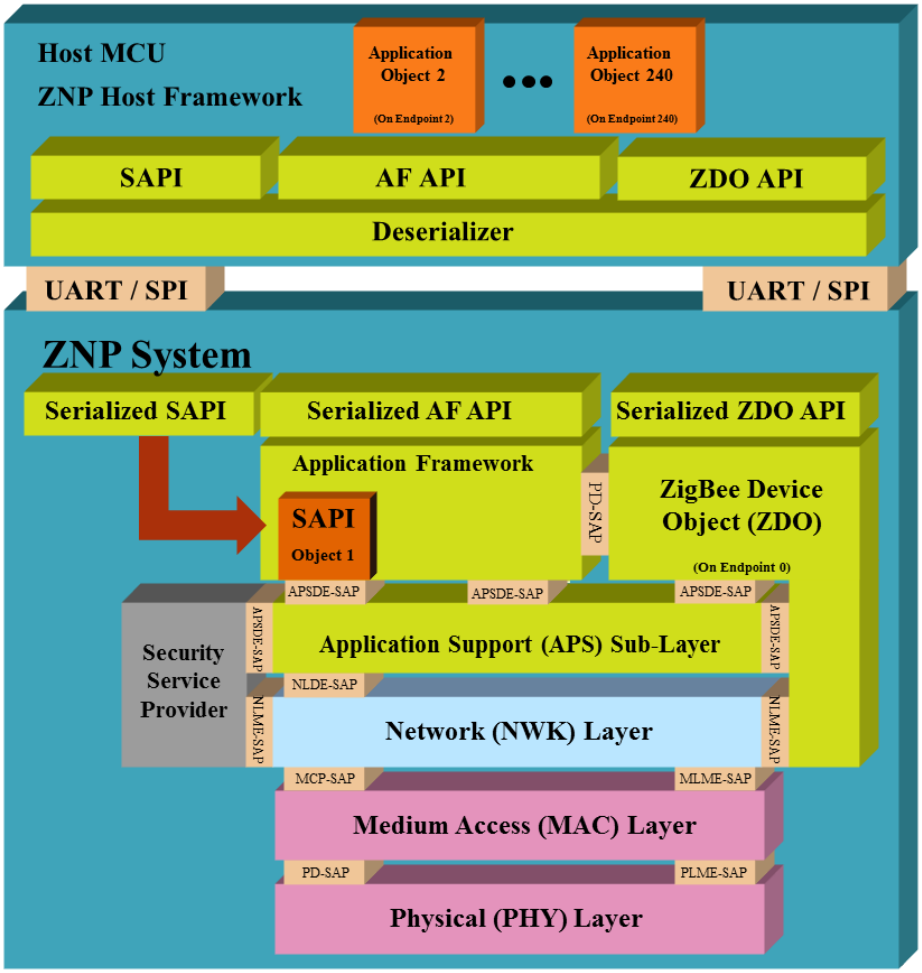


Figure 1. ZNP Host Framework and ZNP Block Diagram

3.1 Highlighted Products

3.1.1 CC2538EM

The CC2538 device integrates an IEEE 802.15.4 (2.4 GHz) radio, ARM® Cortex-M3® processor, security acceleration, and enough flash and random access memory (RAM) to run the high-memory footprint stack and applications (Figure 2).

The CC2538 also includes hardware support for error correction coding (ECC) and RSA public key calculations. The CC2538 device family currently supports TI's ZigBee PRO stack Z-Stack™ with associated profiles and is suitable for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) and wireless highway addressable remote transducer (HART) implementations. The CC2538 is suited for systems that require very low power consumption. Very low-power sleep modes are available. Short transition times between operating modes further enable low power consumption. For a complete feature list of any of the devices, see the corresponding data sheet.

Figure 2. CC2538EM

3.1.2 Tiva LaunchPad

The Tiva™ C Series TM4C1294 Connected LaunchPad Evaluation Board (EK-TM4C1294XL) is a low-cost evaluation platform for ARM Cortex-M4F-based microcontrollers (Figure 3). The LaunchPad design highlights the TM4C1294NCPDT microcontroller with its on-chip 10/100 Ethernet media access control (MAC) and physical layer (PHY), USB 2.0, hibernation module, motion control pulse-width modulation (PWM), and a multitude of simultaneous serial connectivity. The Connected LaunchPad also features two user switches, four user light-emitting diodes (LEDs), dedicated reset and wake switches, a breadboard expansion option, and two independent BoosterPack XL expansion connectors. The pre-programmed quick start application on the LaunchPad also enables remote monitoring and control of the evaluation board from an internet browser anywhere in the world. The web interface is provided by a third party, Exosite. Each LaunchPad is enabled on the Exosite platform allowing users to create and customizable Internet-of-Things (IoT) applications.

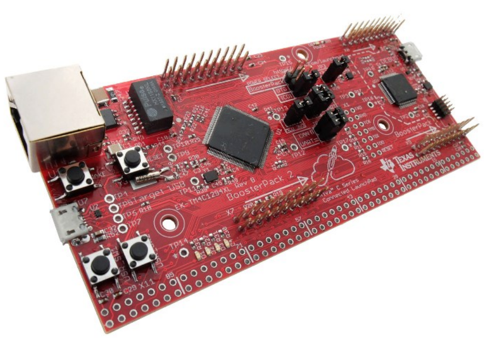


Figure 3. Tiva Connected LaunchPad

4 System Design Theory

A ZNP device must be connected to a host processor through a universal asynchronous receiver/transmitter (UART) or serial peripheral interface (SPI) to use the framework. The host processor is able to control the ZNP by using the ZNP host framework, which provides the ability to communicate with the ZigBee processor. These communications are simplified into an easy-to-use API and callback functions. The host framework is composed of the following abstraction layers:

(a) Remote procedure call (RPC) or transport layer

(b) Z-Stacks monitor and test (MT) subsystems API

(c) Application specific files

Each one of these layers is carefully designed not only to be easy to use but also to have great portability across different operating systems and platforms.

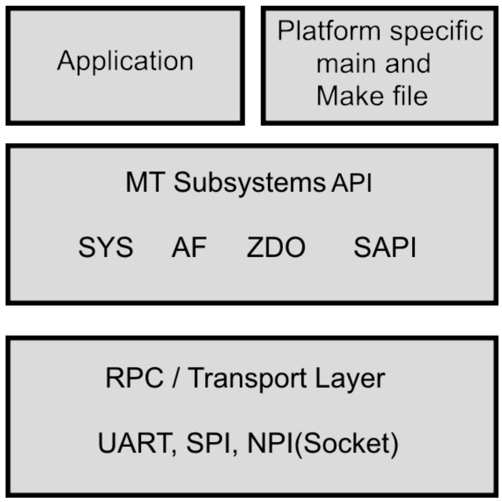


Figure 4. Framework Layers

4.1 RPC or Transport

This layer provides an API together with UART and SPI drivers for the host processor to exchange serial data with the ZNP device.

4.2 MT Subsystems

The MT subsystems layer provides an API for the MT_AF, MT_ZDO, MT_SYS, and MT_SAPI subsystems. This API provides access to all ZNP commands by using callable functions for outgoing commands and callback registration for incoming commands.

4.3 Application and Platform Specific

The application layer contains the callback functions for incoming commands, helper functions, and the appProcess function where the application is developed. The appProcess function is called by a separate thread in the platform-specific main function.

5 Getting Started Hardware

To get started with the ZNP host framework it is necessary to have a POSIX compliant platform such as Linux or a Tiva LaunchPad to use as the host processor and a CC253xEM ZNP.

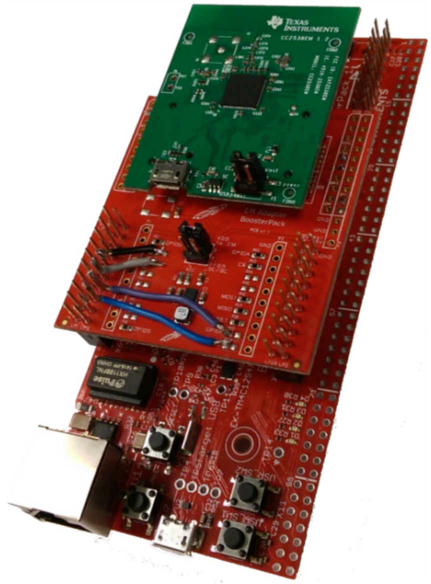


Figure 5. Example of Tiva LaunchPad as a Host

5.1 Host Hardware

The host platforms supported by the ZNP host framework are:

- POSIX compliant platforms.
- TI-RTOS platforms such as the Tiva LaunchPad. The Tiva LaunchPad requires a booster pack in order to connect with a ZNP device such as a CC2538EM.
  - Tiva LaunchPad: https://store.ti.com/tiva-connected-launchpad.aspx
  - EM Adapter BoosterPack: https://store.ti.com/boost-ccemadapter.aspx

5.2 ZNP Hardware

The ZNP host framework supports any CC253x ZNP platform. This document focuses on the CC2538 ZNP and SmartRF06 board, which are required to flash the CC2538EM.

5.3 Modifying the BoosterPack

The BoosterPack requires modifying to allow the UART flow control to be used. The CCxxxxEM adapter BoosterPack connects the Tiva processors UART7 to the CCxxxxEM UART (the UART 7 of the Tiva does not support flow control). The following modifications (Figure 6 and Figure 7) connect to the Tiva UART 4 Tx, Rx, CTS, and RTS on pins K0, K1, K2, and K3 to the UART port on the CCxxxxEM UART.

The following steps show how to make the required hardware modifications:

1. Remove the R3, R4, and R7 resistors (Figure 6).

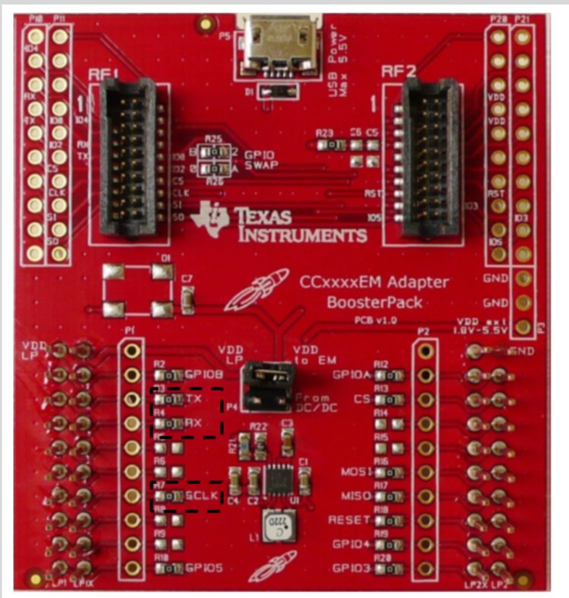


Figure 6. Resistors to be Removed

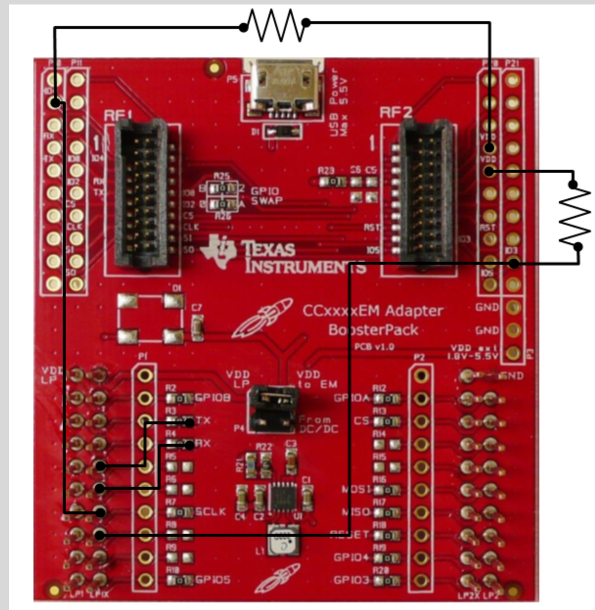


Figure 7. Diagram for New Connections

The following steps are illustrated in Figure 7:

2. Connect LP1X pin 5 to TX.
3. Connect LP1X pin 6 to RX.
4. Connect LP1X pin 7 to P11 pin 7.
5. Connect LP1X pin 8 to P21 pin 9.
6. Connect a pull-up 10-kΩ resistor between P11 pin 7 and P20 pin 4.
7. Connect a pull-up 10-kΩ resistor between P21 pin 9 and P20 pin 5.

6 Getting Started Firmware

The first step to working with the ZNP host framework is to download the following required software tools:

(a) SmartRF™ Flash Programmer 2: Software for programming the ZNP hex files (http://www.ti.com/tool/flash-programmer).

(b) Code Composer Studio™: Software that provides a tool for compiling, linking, debugging, and loading applications on TI-RTOS devices (http://processors.wiki.ti.com/index.php/Download_CCS). This software is not required for Linux platform builds.

6.1 Downloading the ZNP Host Framework

The ZNP host framework project is at https://git.ti.com/znp-host-framework/znp-host-framework. Create an account on the Gitorious@TI Open Source Collaboration website in order to clone this repository (https://git.ti.com).

6.2 Programming the CC2538 With the ZNP Firmware

Fit the CC2538 evaluation module on the SmartRF06. Ensure that the P5 header of the CC2538EM is set to the default position connecting the pin VDD to the EB power pin, so that the SmartRF06 powers the evaluation module.

Connect the SmartRF06 USB to a PC and open the SmartRF Flash Programmer 2 software. In SmartRF Flash Programmer 2:

1. Click the *Refresh* button to discover the CC2538.
2. Select the CC2538 under *Connected devices*.
3. Browse the ZNP hex image provided in the ZNP host framework folder (under the *Main* tab and *Flash image(s)* heading).
4. Select the radial button for *Exclude in image filled with:* (under the *Program*action tab). Enter 0 into the *Exclude in image filled with:* field.
5. Check the radial boxes for the *Erase*, *Program*, and *Verify* actions.

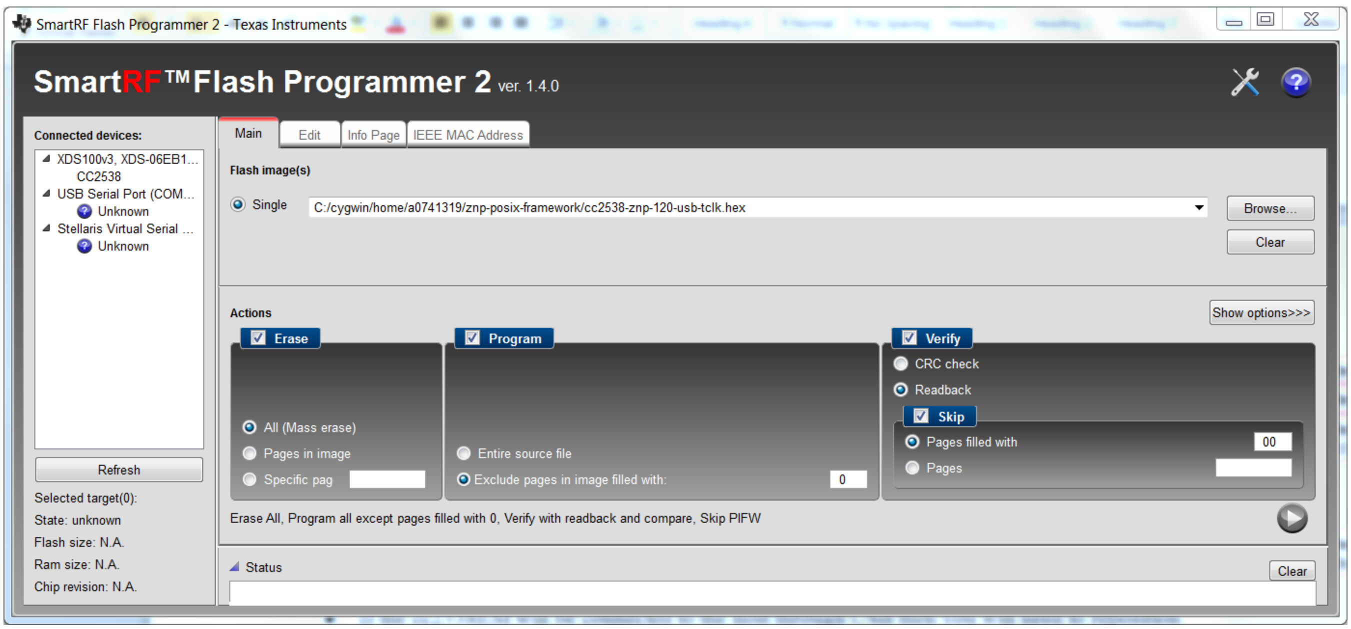


Figure 8. SmartRF Programmer 2

The CC2538EM must be detached from the SmartRF06 board after the firmware image is flashed. If the CC2538EM connects to the host through a USB then the user must reposition the jumper link to the position shown in red in Figure 9.

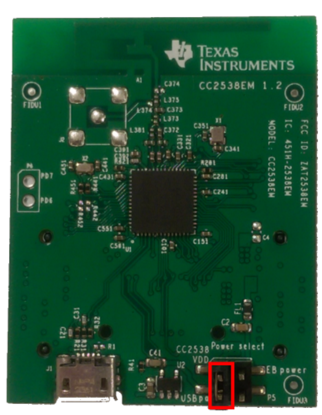


Figure 9. Jumper Position for USB Connection

6.3 Running the ZNP Host Framework on Linux

After flashing the CC2538EM with the ZNP hex image, make sure the jumper on P5 is in the position shown in Figure 10 and plug the CC2538EM into a Linux computer.

To compile the examples, open a terminal window and navigate to the directory where the example is located, such as “<Installation Directory>/examples/dataSendRcv/build/gnu” when using a Linux platform. While in this directory, type in *make* to compile the example.

If the example has already been compiled, then enter the following command in the same directory:

```
sudo  ./<Executable> <Port assigned to ZNP Device>
Example:
sudo  ./dataSendRcv.bin /dev/ttyACM0
```

6.4 Running the ZNP Host Framework on TI-RTOS

To use the examples with TI-RTOS, follow these steps:

1. Open Code Composer Studio.
2. Go to the menu *View* and click on *CCS App Center* (Figure 10). Once the App Center is open, type "TI-RTOS" in the search bar.

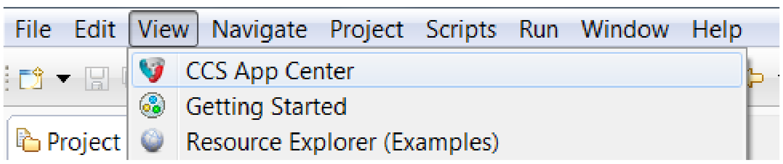


Figure 10. Opening App Center

3. Select and download TI-RTOS for Tiva C.

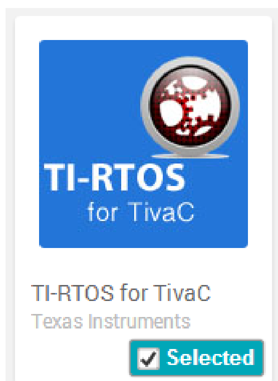


Figure 11. TI-RTOS CCS Add-On

4. After installing TI-RTOS, go to the following link to download and install TI-RTOS 2.00.02.36 for Tiva C (http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/tirtos/index.html).

5. Restart CCS after installing TI-RTOS 2.00.02.36.
6. Now that the environment is set up, the user must import the desired example to run. To import the project, go to *File* then *Import*. In the Import window, click the *Code Composer Studio* folder to expand the options and click on *CCS Projects*—then click on next (Figure 12).

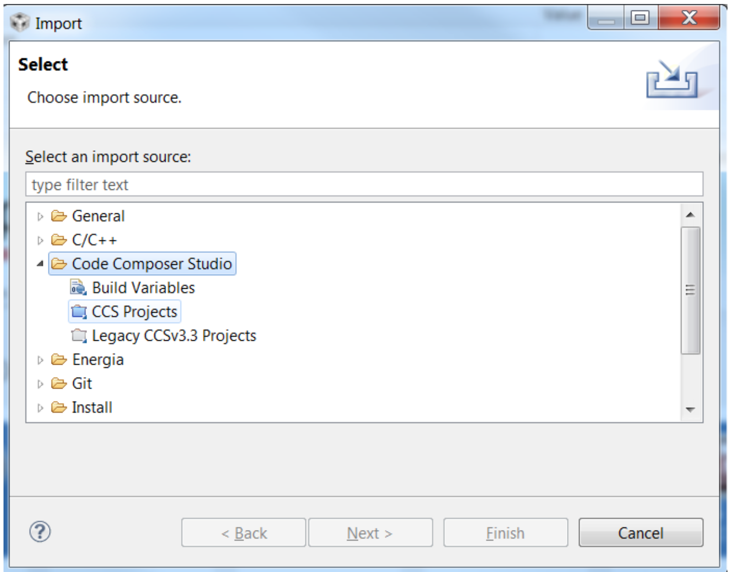


Figure 12. Import Window

7. Click the radial button for *Select search-directory:*, then click the *Browse* button and go to the path <framework installation dir>/examples/Name of Example>. Click *Ok* and make sure that both check box options at the bottom of the window are checked, then click the *Finish* button.

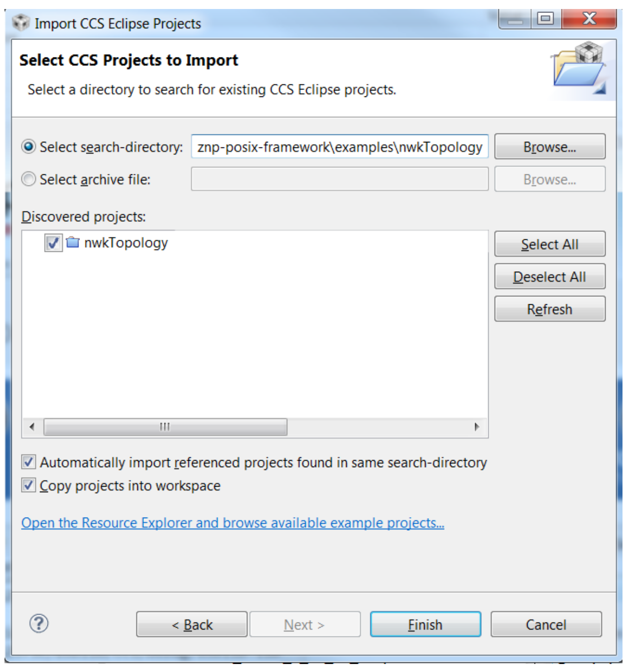


Figure 13. Import Options

8. Now connect the BoosterPack and CC2538EM to the Tiva LaunchPad, which then connects to the computer.
9. Using the serial console of whatever individual preference (such as PuTTy), open the serial port assigned to the LaunchPad. After opening the serial port, go to CCS and click the Debug button followed by the *Resume* button (Figure 14 and Figure 15).

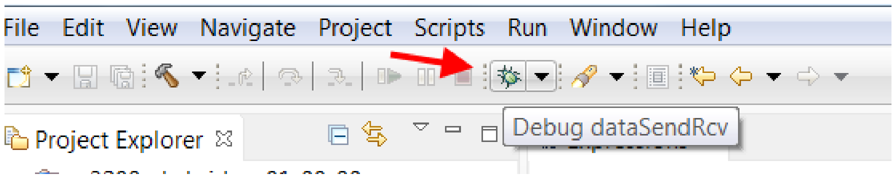


Figure 14. Debug

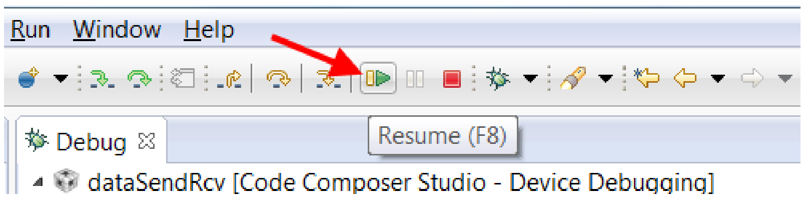


Figure 15. Resume

10. Proceed to the serial console to use the example.

6.5 Examples

Please note, the console allows the user to use the backspace key, which is particularly useful in the command line example. However, on the TI RTOS platform, the console is accessed through a terminal emulator, which requires configuration to correctly send a backspace key press. Configure this backspace key press in the Tera Term program by navigating the menu to *Setup* and then *Keyboard*: (Figure 16).

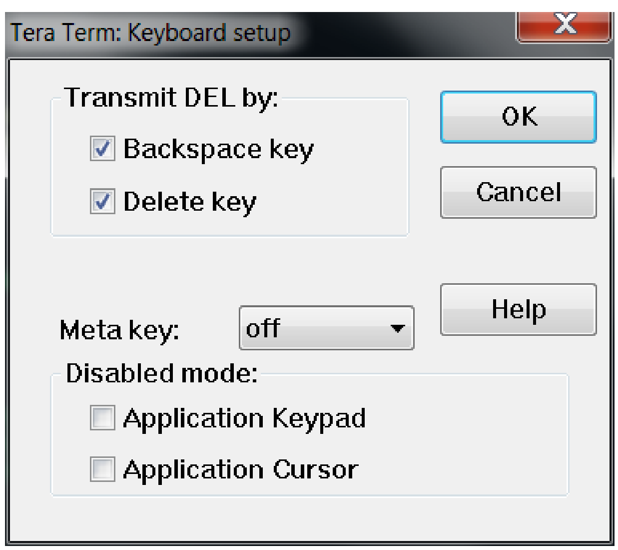


Figure 16. Terminal Emulator Backspace Settings

Ensure that the *Backspace key* check box is selected. Other terminal emulators may have similar settings that require configuring to make the *backspace key* work correctly .

6.5.1 Creating or Joining a New Network

Whenever running any of the examples in the library, the program prompts the user to start or join a new network. If the ZNP device in use has not joined a network (or starting a new network is desired), then type "y" and press enter (Figure 17).

```
Do you want to start/join a new network? (y/n)
y
```

Figure 17. New Network Prompt

Starting a new network resets the configuration in the ZNP. The program then prompts to select what type of device the user desires the ZNP to be: coordinator, router, or end device (Figure 18).

```
Resetting ZNP

ZNP Version: 2.6.1
Enter device type c: Coordinator, r: Router, e: End Device:
c
```

Figure 18. Device Type

After setting the device type, the user must enter the desired channel in which to operate the device. The device then either starts or joins a new network in the chosen channel (Figure 19).

```
Resetting ZNP

ZNP Version: 2.6.1
Enter device type c: Coordinator, r: Router, e: End Device:
c
```

Figure 19. Channel Selection

6.5.2 Command Line Trainer

After setting up the network, the user can enter the desired commands to send. Some of the features of this example are:

- Press the *tab* key twice to see all of the available commands.
- Press the *tab* key once to autocomplete the command that is being entered.
- Press the *enter* key, without entering a command, to display any incoming messages in queue.
- Use the *up* and *down* arrow keys to see the history of commands previously entered.
- After typing the full command name, press the *tab* key twice to display a description of the command.
- Press the *enter* key after typing the command name to select the command and fill out the values for the parameters.

```
Enter CMD
ZDO_IEEE_ADDR_REQ
Description:
 This command will request a device's IEEE 64-bit address.

Command: ZDO_IEEE_ADDR_REQ
Enter ShortAddr: (2B)
0x0000
Enter ReqType: (1B)
1
Enter StartIndex: (1B)
0
mtZdoIeeeAddrRspCb
Status: 0x00
IEEEAddr: 0x040EF65000124B00
NwkAddr: 0x0000
StartIndex: 0x00
NumAssocDev: 0x00
```

Figure 20. Command Example

Color code for the text types in Figure 20:

- **White**: User input
- **Green**: Help (press the tab key twice after typing a command)
- **Blue**: Parameter request
- **Yellow**: Incoming messages from ZNP

6.5.3 Data Send and Receive

After setting up the network, a list of available addresses and endpoints to send messages to is displayed.

```
Available addresses:
Type: COORDINATOR
NwkAddr: 0x0000
Number of Endpoints: 1
Active Endpoints: 0x01

Type: ROUTER
NwkAddr: 0xBF1D
Number of Endpoints: 1
Active Endpoints: 0x01
```

Figure 21. List of Available Devices

Fill in the destination address and the destination endpoint of the targeted device for exchanging messages.

```
Enter DstAddr:
BF1D
Enter DstEndpoint:
1
```

Figure 22. Destination Address and Endpoint of Destination

Upon completing the previous step, the user is able to exchange messages with the selected device. To change the destination of the messages, type “CHANGE” to select a different device as a destination or “QUIT” to terminate the program.

```
Enter message to send or type CHANGE to change the destination
or QUIT to exit:
Hello World!


Message transmited Succesfully!
```

Figure 23. Example of Message Transmission

6.5.4 Network Topology

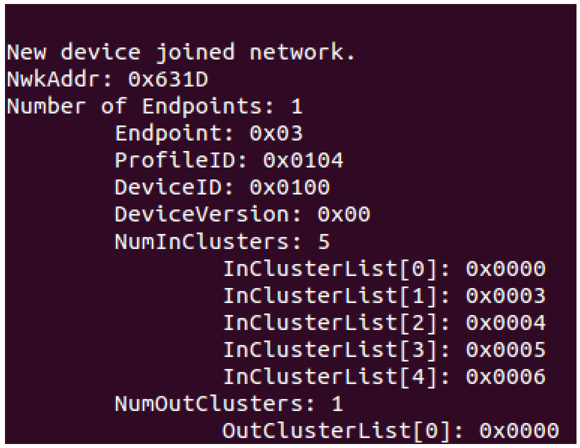
After the network is set up, press the *Enter* key to display the network topology.

```
Press Enter to discover Network Topology:

Node Address: 0x0000   Type: COORDINATOR
Children: 2
        Address: 0x2062   Type: ROUTER
        Address: 0xD6A8   Type: ROUTER

Node Address: 0x2062   Type: ROUTER
Children: 0

Node Address: 0xD6A8   Type: ROUTER
Children: 1
        Address: 0x484B   Type: END DEVICE
```

Figure 24. Network Topology Example

6.5.5 Service Discovery

Run the *serviceDisc* example; wait for a device to join the network and the description of the new device displays automatically.



```
New device joined network.
NwkAddr: 0x631D
Number of Endpoints: 1
        Endpoint: 0x03
        ProfileID: 0x0104
        DeviceID: 0x0100
        DeviceVersion: 0x00
        NumInClusters: 5
                InClusterList[0]: 0x0000
                InClusterList[1]: 0x0003
                InClusterList[2]: 0x0004
                InClusterList[3]: 0x0005
                InClusterList[4]: 0x0006
        NumOutClusters: 1
                OutClusterList[0]: 0x0000
```

Figure 25. Service Discovery Example

7 Test Setup

To set up a test environment it is necessary to have two ZNP devices and two hosts running the ZNP host framework example *stressTest* (found in the examples folder). On the user interface, set one device as a router and the other device as a coordinator and configure both devices to the same channel. Then the router pairs with the coordinator and the automated test begins.

8 Test Data

TEST CASE	EXAMPLE	DUT CONFIGURATION					RESULT	COMMENTS
		LINUX	TI RTOS	COORD	ROUTER	END DEVICE		
Stress test Tiva Coord	stressTest		x	x			Pass	Pass Cnt: x Error Cnt: y ED not tested (see limitations in ReadMe.MD)
	stressTest	x			x			
Stress test Linux Coord	stressTest	x		x			Pass	Pass Cnt: x Error Cnt: y ED not tested (see limitations in ReadMe.MD)
	stressTest	x			x			
	stressTest		x		x			
Command Line	cmdLine	x		x			Pass	
	cmdLine	x			x		Pass	
	cmdLine	x				x	Pass	
	cmdLine		x		x		Pass	
Data Send Receive (Linux Coord)	dataSendRecv	x		x			Pass	
	dataSendRecv	x			x		Pass	
	dataSendRecv	x				x	Pass	
	dataSendRecv		x			x	Pass	
Data Send Receive (TI RTOS Coord)	dataSendRecv		x	x			Pass	
	dataSendRecv	x			x		Pass	
	dataSendRecv	x				x	Pass	
	dataSendRecv		x		x	x	Pass	
Network Topology (Linux Coord)	nwkTopology	x		x			Pass	
	nwkTopology	x			x		Pass	
	nwkTopology	x				x	Pass	
	nwkTopology		x			x	Pass	

9 Design Files

9.1 Schematics

To download the schematics for each board, see the design files at TIDC-ZNP-HOST-SW3.

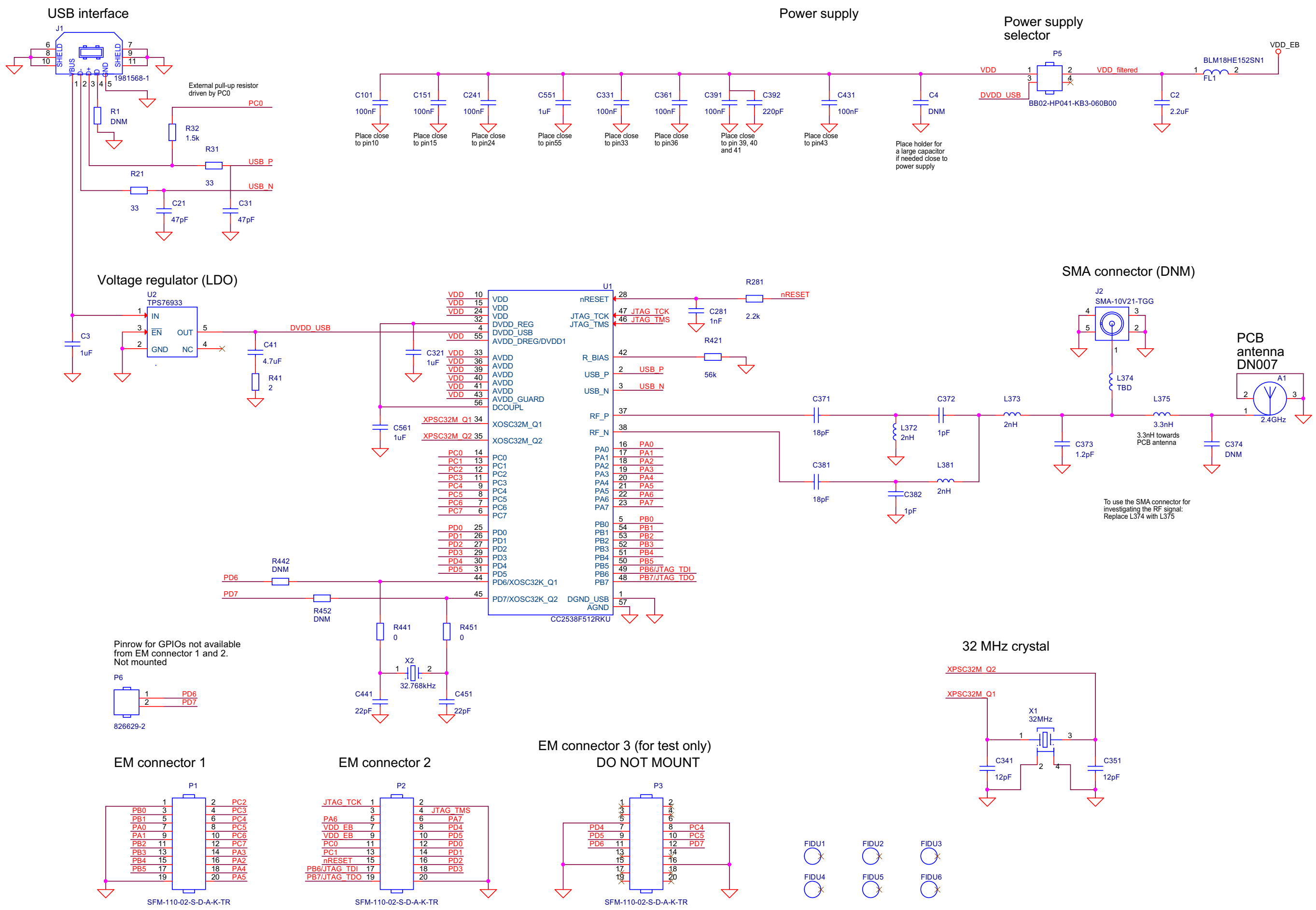


Figure 26. TIDC-ZNP-HOST-SW3 Schematics

9.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDC-ZNP-HOST-SW3.

9.3 Layer Plots

To download the layer plots, see the design files at TIDC-ZNP-HOST-SW3.

9.4 Altium Project

To download the Altium project files, see the design files at TIDC-ZNP-HOST-SW3..

9.5 Layout Guidelines

To download the Layer files, see the design files at TIDC-ZNP-HOST-SW3.

9.6 Gerber Files

To download the Gerber files, see the design files at TIDC-ZNP-HOST-SW3.

9.7 Assembly Drawings

To download the Assembly- files, see the design files at TIDC-ZNP-HOST-SW3.

9.8 Software Files

To download the software files, see the design files at TIDC-ZNP-HOST-SW3.

10 References

1. Texas Instruments. "*CC2538ZNP Interface Specification*" Application Report within the *Z-STACK-HOME: ZigBee Home Automation Solutions* Software Application, (http://www.ti.com/tool/z-stack).
   - This document can be found in the installation directory of the Z-Stack Home under the following path {user install directory}\Z-Stack Home 1.2.1\Documents\API\CC2538\CC2538ZNP Interface Specification.pdf. Visit http://www.ti.com/tool/z-stack to install the Z-STACK-HOME: ZigBee Home Automation Solutions software and access the reference material.

11 Terminology

AF— ZigBee Application Framework

API— Application Programming interface

MCU— Micro Controller Unit

MPU— Micro Processor Unit

MT— Z-Stack's Monitor and Test Layer

NPI— Network Peripheral Interface

RPC— Remote Procedure Call

SAPI— Simple Application Programming interface

SPI— Serial Peripheral Interface bus

UART— Universal Asynchronous Receiver Transmitter

ZC— ZigBee Coordinator

ZDO— ZigBee Device Object

ZED— ZigBee End Device

ZNP— ZigBee Network Processor

ZR— ZigBee Router

12 About the Author

HECTOR RAMOS is an Applications Engineer at Texas Instruments, where he is responsible for supporting costumers and developing software applications. Hector earned his Bachelors of Science in Electrical Engineering and Computer Science from University of California Berkeley, where he acquired various skills on software development, analog and digital circuit design, and integrated circuits design.

TONY CAVE has been working for Texas Instruments as a field and factory application engineer for 10 years. In the last 4 years he specialized in 802.15.4, mesh protocols, ZigBee technology and cloud service integration. Tony has more than 15 years of experience in Embedded SW Development and earned his Bachelors of Science from the Open University England, where he acquired skills in Digital and Analogue Electronic Engineering and Communication Protocols.

IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

Texas Instruments Incorporated ("TI") reference designs are solely intended to assist designers (“Buyers”) who are developing systems that incorporate TI semiconductor products (also referred to herein as “components”). Buyer understands and agrees that Buyer remains responsible for using its independent analysis, evaluation and judgment in designing Buyer’s systems and products.

TI reference designs have been created using standard laboratory conditions and engineering practices. **TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.** TI may make corrections, enhancements, improvements and other changes to its reference designs.

Buyers are authorized to use TI reference designs with the TI component(s) identified in each particular reference design and to modify the reference design in the development of their end products. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY THIRD PARTY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT, IS GRANTED HEREIN, including but not limited to 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.

TI REFERENCE DESIGNS ARE PROVIDED "AS IS". TI MAKES NO WARRANTIES OR REPRESENTATIONS WITH REGARD TO THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING ACCURACY OR COMPLETENESS. TI DISCLAIMS ANY WARRANTY OF TITLE AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT, QUIET POSSESSION, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO TI REFERENCE DESIGNS OR USE THEREOF. TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY BUYERS AGAINST ANY THIRD PARTY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON A COMBINATION OF COMPONENTS PROVIDED IN A TI REFERENCE DESIGN. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY AND WHETHER OR NOT TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT OF TI REFERENCE DESIGNS OR BUYER’S USE OF TI REFERENCE DESIGNS.

TI reserves 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 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 for TI components 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.

Reproduction of significant portions of TI information in TI data books, data sheets or reference designs 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.

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 that anticipate dangerous failures, monitor failures and their consequences, lessen the likelihood of dangerous failures 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 Buyer’s 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 an agreement specifically governing such use.

Only those TI components that 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 that have ***not*** been so designated is solely at Buyer's risk, and 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.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265