RemoTI^{TM} Win32 SimpleConsole Application

Keywords

- *RemoTI^{TM}*
- *ZigBee RF4CE*
- *RTI PC dll*
- *RemoTI Network Processor*
- *Visual Studio C++ 2010 Express Edition*
- *CC2533 / CC2530*

Introduction

One of the powerful features of RemoTI is the uniform application programming interface across different platform configuration. RemoTI basically presents the same interface if you develop your application on the CC253x only or if you develop your application on an MCU and utilize the C253x as a network processor.

The RemoTI software release also includes a dll to facilitate development in a Win32 PC environment. This dll presents the exact interface as on the embedded device and can therefore be used to develop and test application code before your custom hardware arrives, develop tools etc. The Target Emulator included in the RemoTI release is written using this dll.

This application note describes a simple Win32 console application exercising the dll. The main objective of the application note is to guide users through project environment setup using Visual Studio C++ 2010 Express edition and to provide a very simple target node and remote controller node application to showcase how to initialize, pair and send/receive ZRC and ZID commands.

The application is kept simple and will only highlight the very basic steps needed to create a Win32 console application. Among other things, no GUI software library is used and key presses on the command window are used to synchronize events. Please refer to the sample applications in the RemoTI software release for more complete RemoTI examples. Project collateral discussed in this application note can be downloaded from the following URL: http://www.ti.com/lit/zip/SWRA304.

Table of Contents

Abbreviations

API	Application Programmer Interface
ZRC	Consumer Electronics Remote Control
ZID	ZigBee Input Device
DLL	Dynamically Linked Library
EM	Evaluation module
EB	Evaluation Board
RTI	RemoTI Application Interface
RF4CE	Radio Frequency for Consumer Electronics
SDK	Software Development Kit
TE	Target Emulator
UI	User Interface

1. System Setup

The Win32 SimpleConsole application communicates with a CC253x RemoTI network processor (RNP) using a UART connection. The software included with this application note includes a sample application configuring the RNP as a remote controller (RC) node and another sample application configuring the RNP as a target node.

The Target Emulator (TE) application included with the RemoTI release can be used with the RemoTI kit target board to enable discovery/pairing and display the messages sent from the SimpleConsole application configured as an RC node. See Figure 1. The TE is connected to UART1 and the SimpleConsole application is connected to UART2. Note that the RemoTI kit only includes one target board and you need two target boards to test this setup. The SmartRF05EB+CC253xEM can also be configured as a RNP.

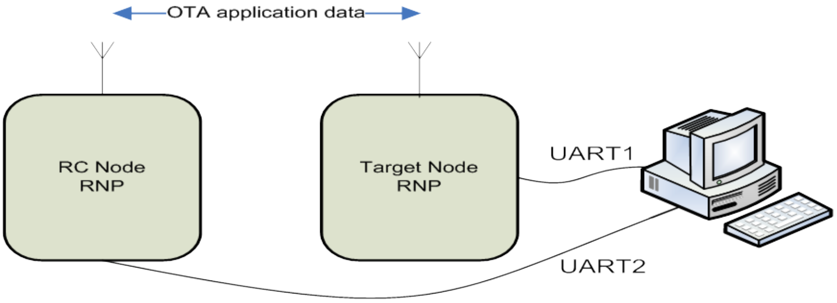


Figure 1. SimpleConsole Application configured as a RC Node

The basic RC included with the RemoTI kit can be used to trigger discovery/pairing and send ZRC command to the SimpleConsole application configured as a target node. See Figure 2.

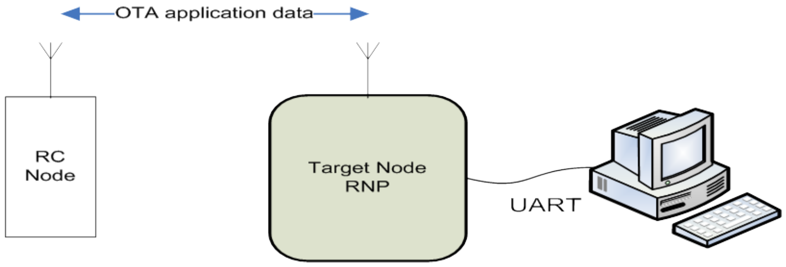


Figure 2. SimpleConsole Application Configured as a Target Node

2. Installation Requirements

2.1 Host Computer Requirements

The RTI dll was tested on a personal computer running Microsoft Windows XP Professional. The following are the minimum platform requirements for using the RTI dll.

- Visual C++ 2010 Express Edition
- Windows 7
- Multiple USB ports to connect to target network processor
- RemoTI 1.3.1 software release (www.ti.com/remoti)

2.2 Target System Requirements

The application developed using RTI dll connects to the RemoTI network processors via a UART interface. It is important that the RemoTI network processor is built with the same version of the software as the RTI dll.

- IAR EW 8051 Embedded Workbench (www.iar.com). The RemoTI software release list the IAR version requirement for the release.
- Depending on the application, an additional RemoTI target Board or SmartRF05EB+CC253xEM may be required.

3. Product Installation Requirements

3.1 Install RemoTI

The default installation folder for RemoTI 1.3.1 this is “C:\Texas Instruments\RemoTI-CC253xDK-1.3.1”. This folder is referenced in the rest of this application note. If you install RemoTI in another folder, you must reference your specific install folder in project setup settings, configuration paths etc that are referenced later in this application note. Note that you also need to upgrade this path when installing future RemoTI releases.

3.2 Install Visual C++ 2010 Express Edition

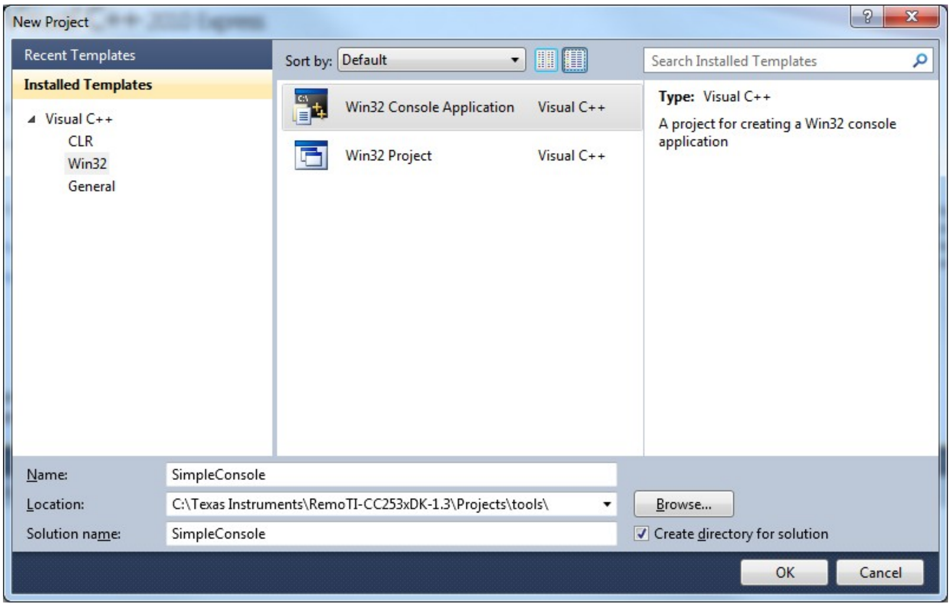
Go to http://www.microsoft.com/express/product and follow the instruction to download and install Visual C++ 2010 Express Edition. Note that other tools, such as Visual C++ 2005/2008 Express Edition and Visual Studio 2005/2008, will work but the configuration and setup instructions may be different from what is captured in this document.

4. Create and Configure the SimpleConsole Application

This section provides the steps necessary to configure a simple RemoTI application using the RTI dll. If you don’t want to go through these steps, you can unzip the sample application included with this application note to C:\Documents and Settings\’user’\My Documents\Visual Studio 2010\Projects, and go to section 6 for instructions for how to execute the application.

4.1 Create a Win32 Console Application

1. From the Visual C++ Express Edition main menu choose “File – New Project”
    a. Select Win32 project type and Win32 Console Application template
    b. Enter name and path of the project. See Figure 3.



Type: Visual C++

A project for creating a Win32 console application

Name: SimpleConsole

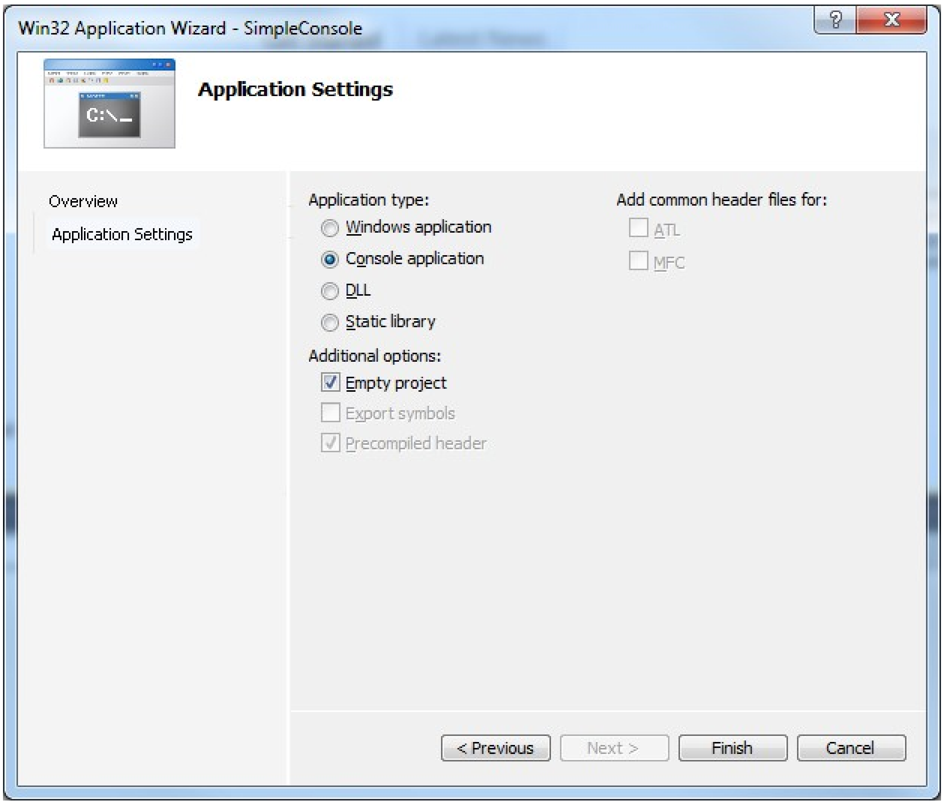
Location: C:\Texas Instruments\RemoTI-CC253xDK-1.3\Projects\tools\

Solution name: SimpleConsole

☑ Create directory for solution

Figure 3. VC++ Express New Project Wizard

2. Click ‘Application Settings’ in the left hand column, and choose empty project in the application wizard. See Figure 4.



Application type:

○ Windows application
● Console application
○ DLL
○ Static library

Additional options:

☑ Empty project
□ Export symbols
☑ Precompiled header

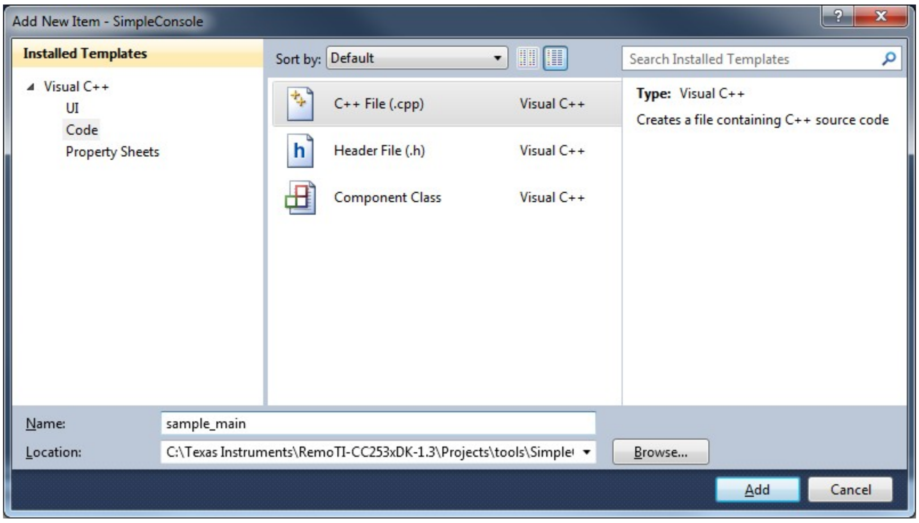
Add common header files for:

□ ATL
□ MFC

Figure 4. VC++ Express Console Application Setting

4.2 Configure Project settings

1. Right click on the “Source Files” folder in the solution explorer and select “Add – New Item”. Select “code” and “C++ File (.cpp)” in the wizard, name the file sample_main and click the add button. See Figure 5.



Add New Item - SimpleConsole

Installed Templates: Visual C++ > UI, Code, Property Sheets

Sort by: Default

C++ File (.cpp)	Visual C++
Header File (.h)	Visual C++
Component Class	Visual C++

Type: Visual C++

Creates a file containing C++ source code

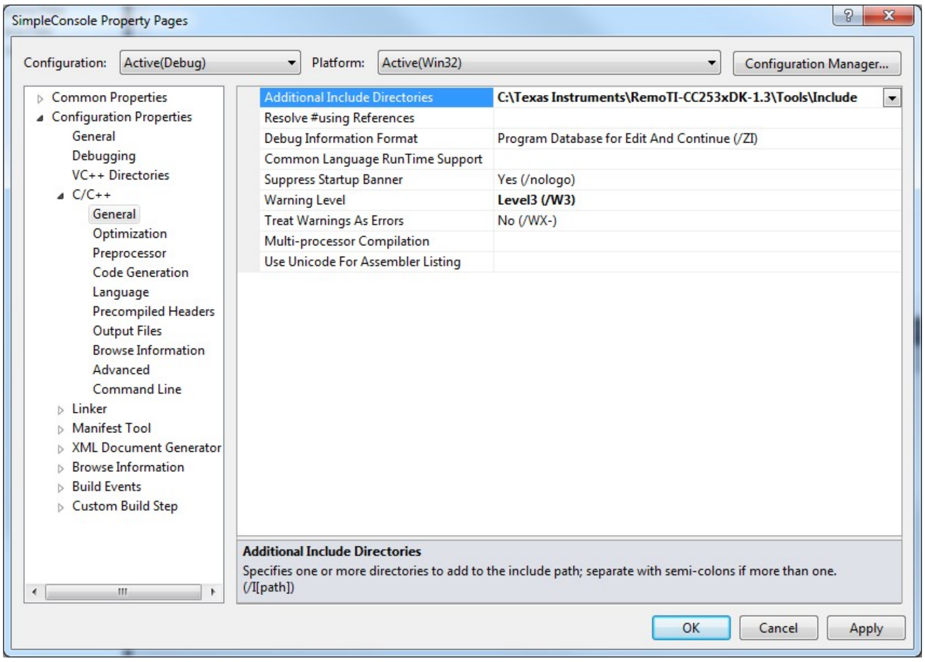
Name: sample_main

Location: C:\Texas Instruments\RemoTI-CC253xDK-1.3\Projects\tools\Simple

Figure 5. VC++ Express Add New Source File Wizard

2. Right-click on the project name (in bold) in the solution explorer and select properties.

   a. Go to C/C++ and the general category and add “C:\Texas Instruments\RemoTI-CC253xDK-1.3.1\Tools\Include” to Additional Include Directories. See Figure 6.



SimpleConsole Property Pages

Configuration: Active(Debug)   Platform: Active(Win32)

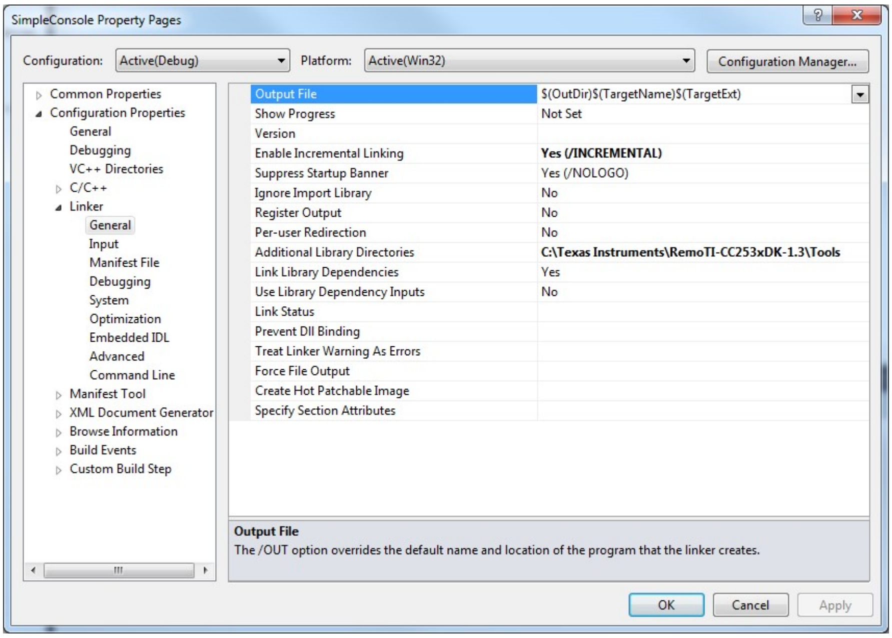
Additional Include Directories	C:\Texas Instruments\RemoTI-CC253xDK-1.3\Tools\Include
Resolve #using References	
Debug Information Format	Program Database for Edit And Continue (/ZI)
Common Language RunTime Support	
Suppress Startup Banner	Yes (/nologo)
Warning Level	Level3 (/W3)
Treat Warnings As Errors	No (/WX-)
Multi-processor Compilation	
Use Unicode For Assembler Listing	

Additional Include Directories

Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[path])

Figure 6. VC++ Express Include Directory Configuration

b. Go to Linker and the General category and add “C:\Texas Instruments\RemoTI-CC253xDK-1.3.1\Tools” to Additional Library Directories. See Figure 7.

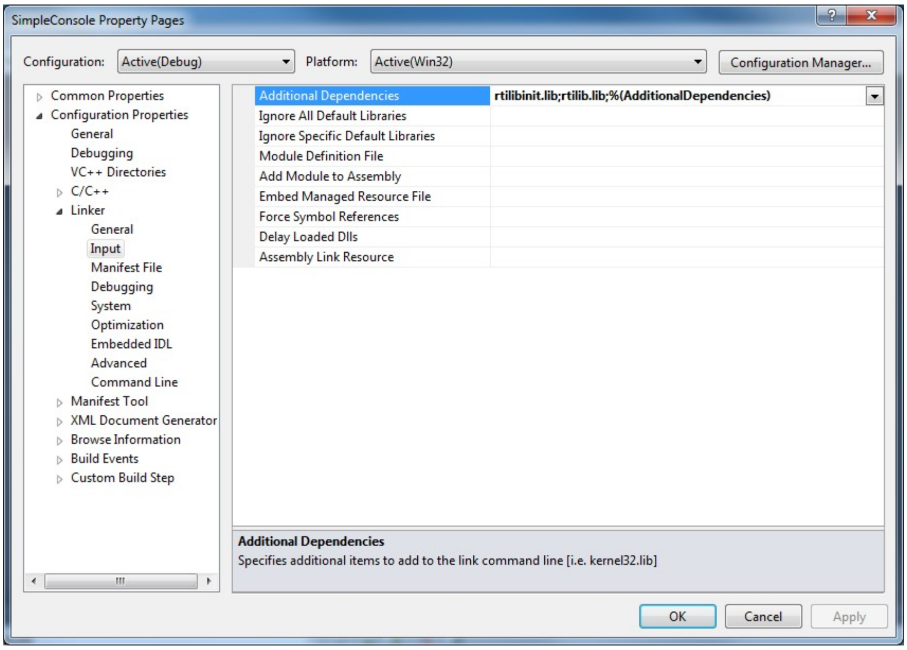


Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	**Yes (/INCREMENTAL)**
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**C:\Texas Instruments\RemoTI-CC253xDK-1.3\Tools**
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

Output File
The /OUT option overrides the default name and location of the program that the linker creates.

Figure 7. VC++ Express Additional Linker Directories Configuration

c. Go to Linker and the Input category and add “rtilibinit.lib; rtilib.lib” (with a ; between the two libraries” to Additional Dependencies. See Figure 8.

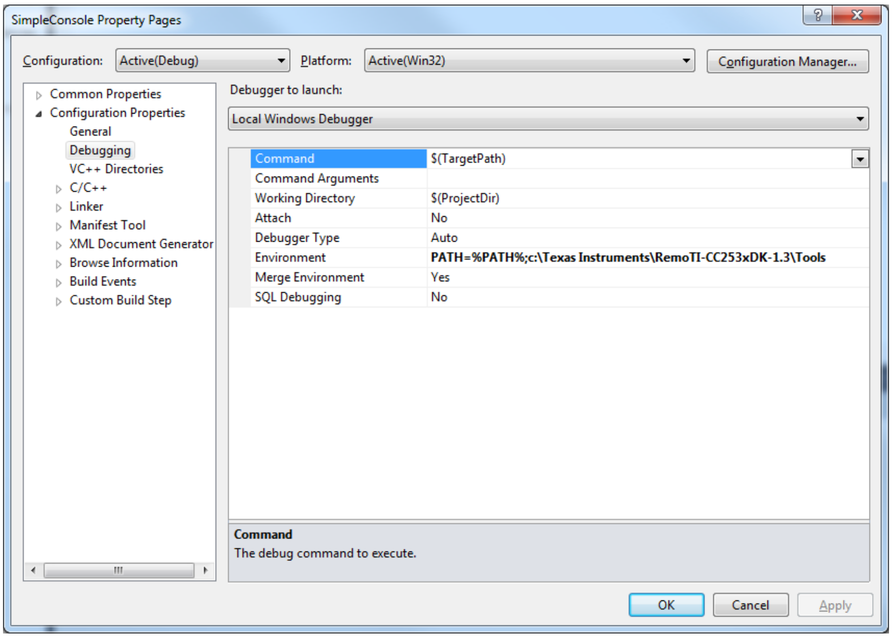


Additional Dependencies	**rtilibinit.lib;rtilib.lib;%(AdditionalDependencies)**
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link command line [i.e. kernel32.lib]

Figure 8. VC++ Express Additional Dependencies Configuration

d. Go to Debugging and the Environment category and add "PATH=%PATH%;c:\Texas Instruments\RemoTI-CC253xDK-1.3.1\Tools". See Figure 9.



SimpleConsole Property Pages

Configuration: Active(Debug)

Platform: Active(Win32)

Debugger to launch: Local Windows Debugger

Command	$(TargetPath)
Command Arguments	
Working Directory	$(ProjectDir)
Attach	No
Debugger Type	Auto
Environment	PATH=%PATH%;c:\Texas Instruments\RemoTI-CC253xDK-1.3\Tools
Merge Environment	Yes
SQL Debugging	No

Command
The debug command to execute.

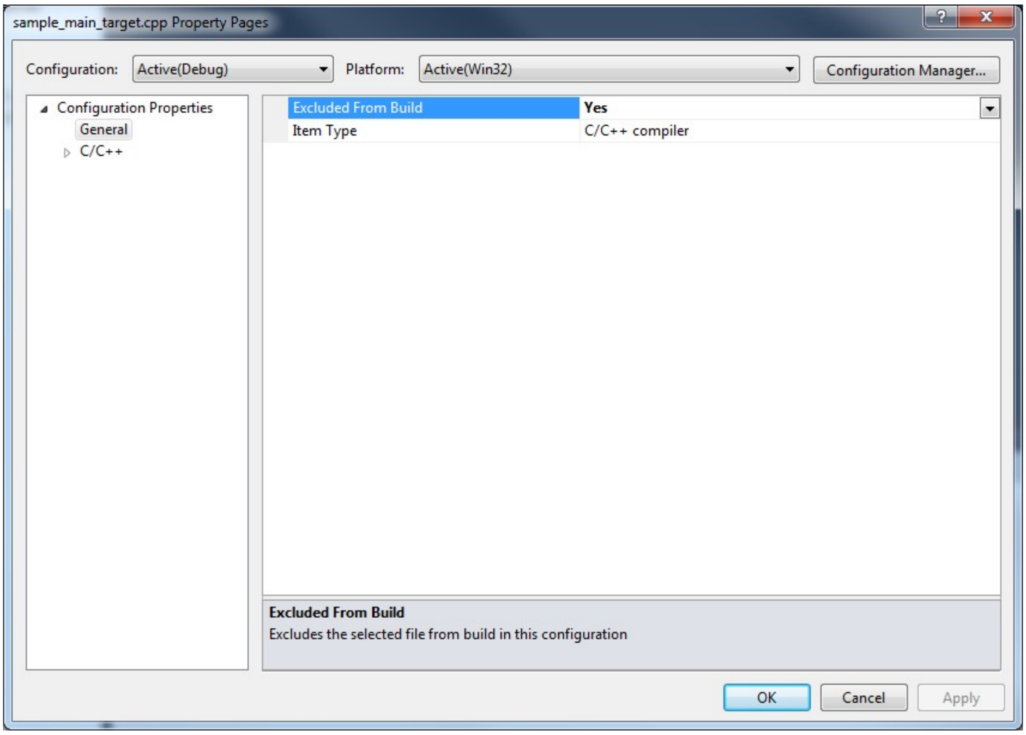
Figure 9. VC++ Express Environment Path Configuration

5. Application Code

We created the file sample_main.c during the configuration process in section 4.2 step 1. This file is still empty but we needed this file as a placeholder in order to complete all the project configurations. The sample code included with this application note contains two main files:

1. sample_main_rc.c. This file contains the application code for running the SimpleConsole application as a RC node.
2. sample_main_target.c. This file contains the application code for running the SimpleConsole application as a target node.

You can copy the content of either of the files above into sample_main.c, but a better approach is to delete the sample_main.c and include both files mentioned above. This way you can include/exclude either of the files for the desired functionality. You can include/exclude a file with a right-click on the file, choose properties and select 'yes' or 'no' in the General -> Exclude from build section as seen in Figure 10.



sample_main_target.cpp Property Pages

Configuration: Active(Debug)
Platform: Active(Win32)

Excluded From Build	Yes
Item Type	C/C++ compiler

Excluded From Build
Excludes the selected file from build in this configuration

Figure 10. VC++ Express File Exclude from Build Configuration

Open up sample_main_rc.cpp or sample_main_target.cpp and notice the RTI APIs called are exactly the same as if the code was developed for the CC253x embedded IAR project. See [1] for more details. However, note that RTI_InitWin32Module() and RTI_CloseWin32Module() are special APIs included in rtilib.dll. These APIs will open and close the connection to the COM port used to communicate with the RNP.

6. Execute the SimpleConsole Application

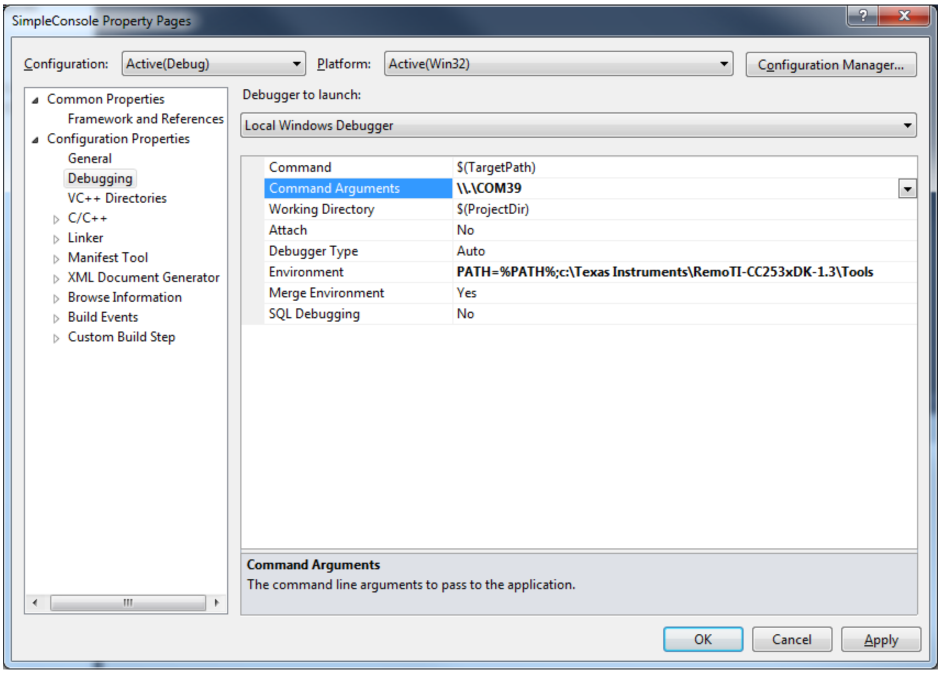
6.1 SimpleConsole Application as RC Node

1. Build the network processor project in C:\Texas Instruments\RemoTI-CC253xDK-1.3.1\Projects\RemoTI\RNP and load the executable onto the target board. Note that

you need to set the correct compiler option depending on the platform you want to use for the RNP for the SimpleConsole application.

   a. If you want to enable ZID on your network processor as Controller; make sure FEATURE_ZID_CLD is set to TRUE under the predefined compileroptions.

2. If you want to enable ZID on your network processor as Target; make sure
3. Connect the mini USB (or RS232) cable between the PC and the target board and go to the PC device manger and note the specific COM port designated for this connection
4. Make sure that sample_main_rc.cpp is included and sample_main_target.cpp is excluded from the project
5. Right-click on the project name (in bold) in the solution explorer and select properties
   a. Go to the Configuration properties and debugging category and enter the COM port number from step 2 in the Command Arguments field. Note that port 1-4 is entered as COMX and port numbers >4 are entered as \\.\COMXX. See example below for COM39.



Configuration: Active(Debug)

Platform: Active(Win32)

Debugger to launch: Local Windows Debugger

Command	$(TargetPath)
Command Arguments	\\.\COM39
Working Directory	$(ProjectDir)
Attach	No
Debugger Type	Auto
Environment	PATH=%PATH%;c:\Texas Instruments\RemoTI-CC253xDK-1.3\Tools
Merge Environment	Yes
SQL Debugging	No

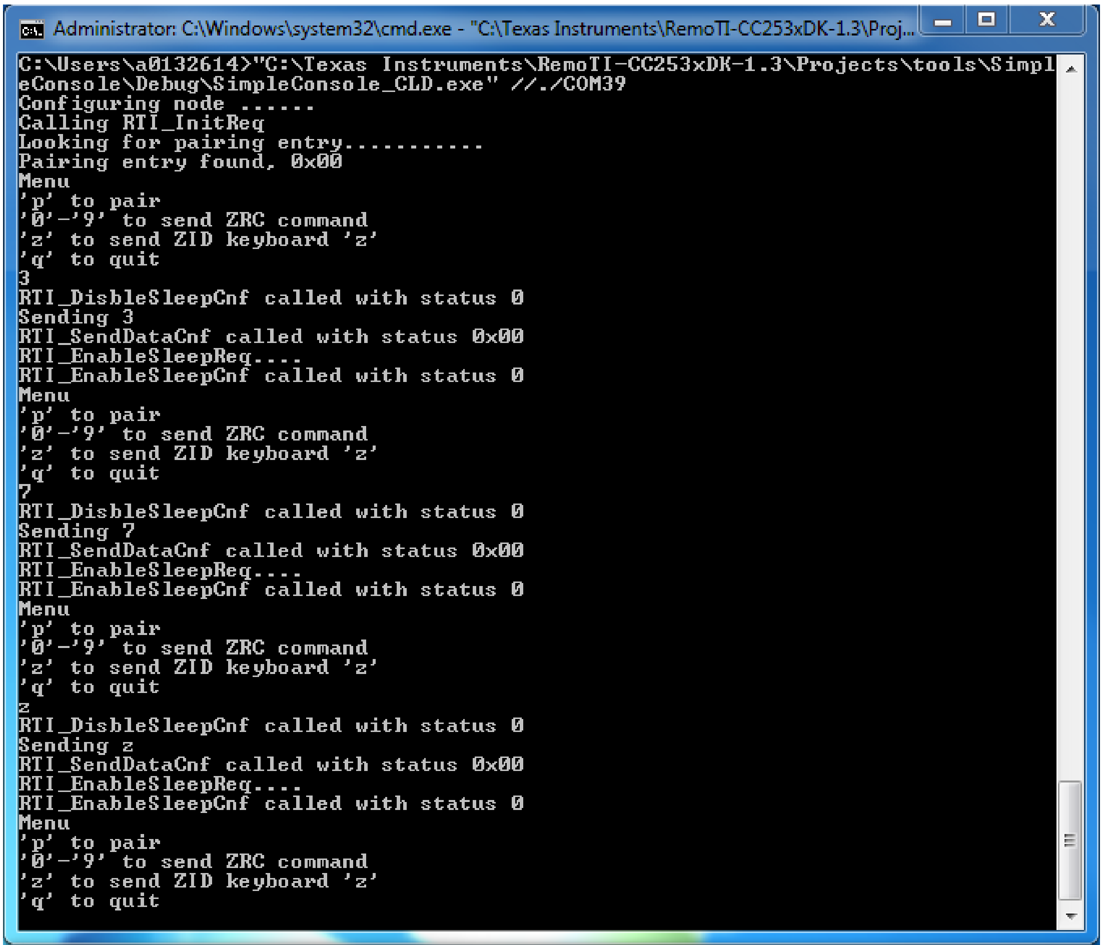
Command Arguments
The command line arguments to pass to the application.

6. Hit the green arrow to start run the application and follow the instruction on the command line to pair and send a ZRC or ZID command over the air. Note that you will need to configure the target emulator on a separate COM port and click the Allow Pair button to enable pairing. After pairing is complete, hit ‘0’-‘9’ to send the corresponding ZRC command to the emulator, ‘z’ to send ZID Keyboard command **‘z’** and ‘q’ to quit the application. When pressing key ‘0’-‘9’, observe the corresponding key highlighted the target emulator.

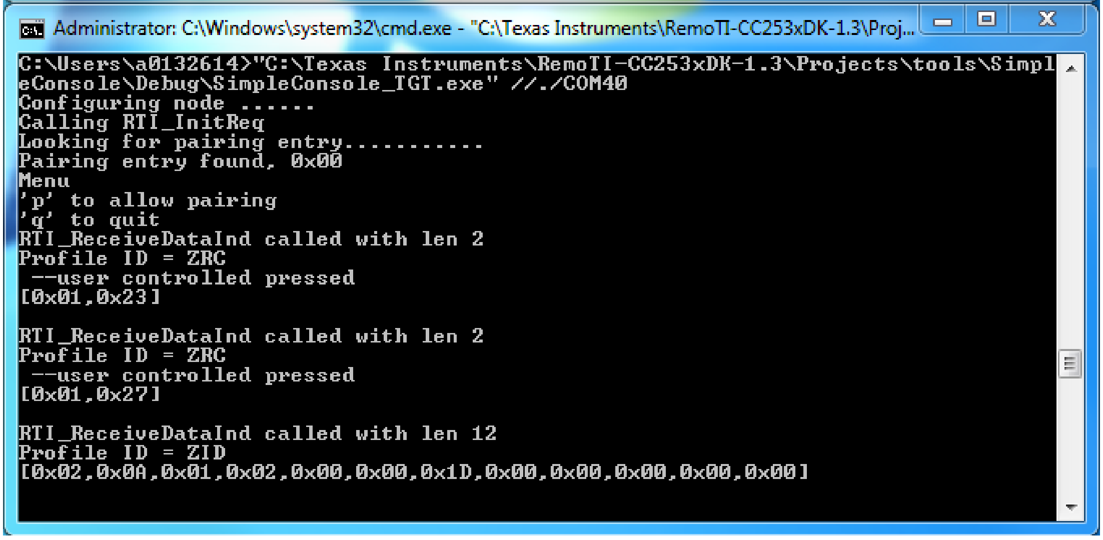
6.2 SimpleConsole Application as Target Node

1. Build the network processor project in C:\Texas Instruments\RemoTI-CC253xDK-1.3.1\Projects\RemoTI\RNP and load the executable onto the targetboard.
   a. FEATURE_ZID_ADA is set to TRUE under the predefined compiler options.
2. Connect the mini USB cable between the PC and the target board and go to the PC device manger and note the specific COM port designated for thisconnection.
3. Make sure that sample_main_target.cpp is included and sample_main_rc.cpp is excluded from the project

4. Configure the COM port as described in step 4 of section 6.1.
5. Hit the green arrow to start run the application and follow the instruction on the command line to allow pairing and receive ZRC and ZID command over the air from a RC. After pairing is complete, the ZRC and ZID command codes will be displayed on the command line. The ZRC and ZID commands will not be parsed since this is beyond the scope of this application note.


Administrator: C:\Windows\system32\cmd.exe - "C:\Texas Instruments\RemoTI-CC253xDK-1.3\Proj...

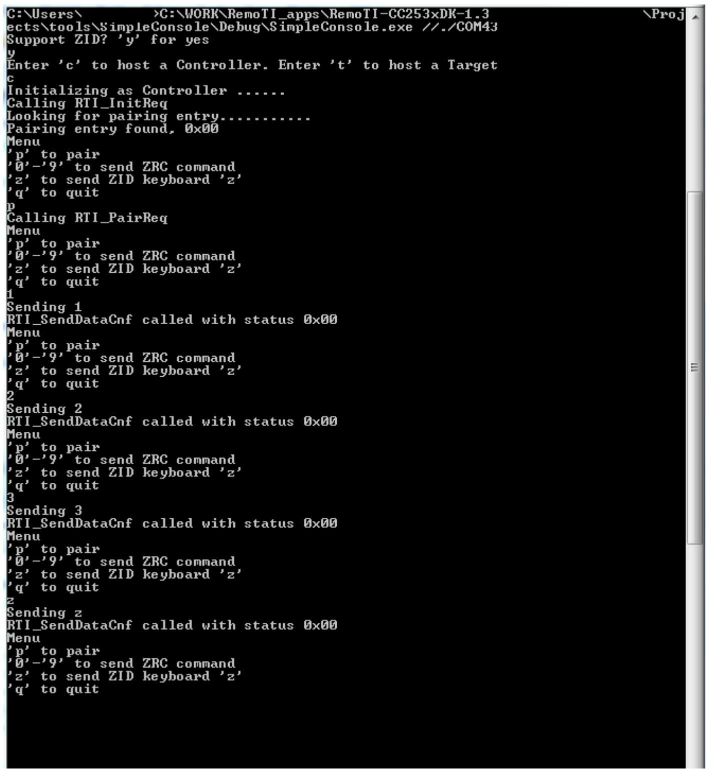
```
C:\Users\a0132614>"C:\Texas Instruments\RemoTI-CC253xDK-1.3\Projects\tools\Simpl
eConsole\Debug\SimpleConsole_CLD.exe" //./COM39
Configuring node ......
Calling RTI_InitReq
Looking for pairing entry...........
Pairing entry found, 0x00
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
3
RTI_DisbleSleepCnf called with status 0
Sending 3
RTI_SendDataCnf called with status 0x00
RTI_EnableSleepReq....
RTI_EnableSleepCnf called with status 0
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
7
RTI_DisbleSleepCnf called with status 0
Sending 7
RTI_SendDataCnf called with status 0x00
RTI_EnableSleepReq....
RTI_EnableSleepCnf called with status 0
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
z
RTI_DisbleSleepCnf called with status 0
Sending z
RTI_SendDataCnf called with status 0x00
RTI_EnableSleepReq....
RTI_EnableSleepCnf called with status 0
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
```


Administrator: C:\Windows\system32\cmd.exe - "C:\Texas Instruments\RemoTI-CC253xDK-1.3\Proj...

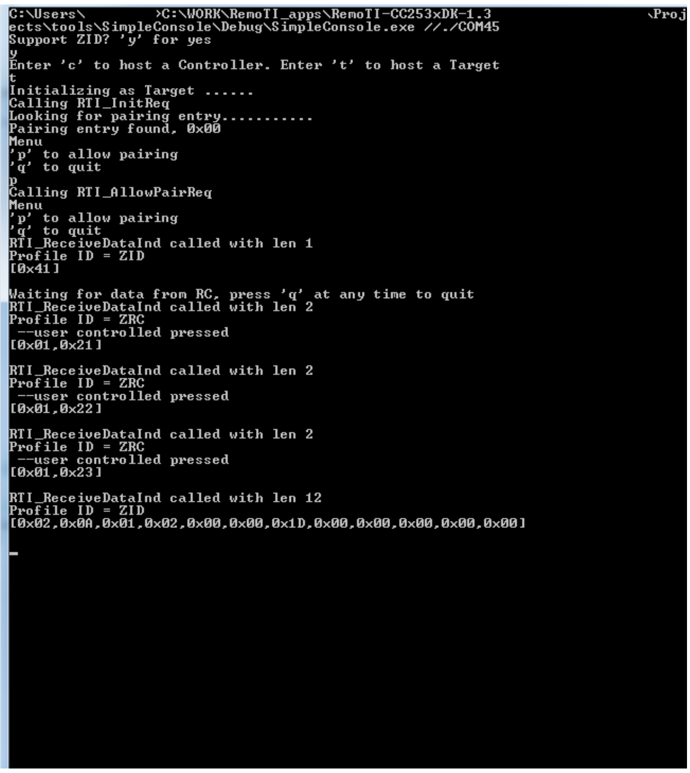
```
C:\Users\a0132614>"C:\Texas Instruments\RemoTI-CC253xDK-1.3\Projects\tools\Simpl
eConsole\Debug\SimpleConsole_TGT.exe" //./COM40
Configuring node ......
Calling RTI_InitReq
Looking for pairing entry...........
Pairing entry found, 0x00
Menu
'p' to allow pairing
'q' to quit
RTI_ReceiveDataInd called with len 2
Profile ID = ZRC
 --user controlled pressed
[0x01,0x23]

RTI_ReceiveDataInd called with len 2
Profile ID = ZRC
 --user controlled pressed
[0x01,0x27]

RTI_ReceiveDataInd called with len 12
Profile ID = ZID
[0x02,0x0A,0x01,0x02,0x00,0x00,0x1D,0x00,0x00,0x00,0x00,0x00]
```

6.3 SimpleConsole Application as either Target or Controller Node

A third configuration exists which combines the two projects described above. Instead of immediately configuration the Target as either Target or Controller this allows the user to choose. It is still a requirement that the RNP supports the configuration that is chosen. Since this combines the functionality of the two configurations the c-code may be more confusing. That is why the two separate projects are kept.



```
C:\Users\         >C:\WORK\RemoTI_apps\RemoTI-CC253xDK-1.3                  \Proj
ects\tools\SimpleConsole\Debug\SimpleConsole.exe //./COM43
Support ZID? 'y' for yes
y
Enter 'c' to host a Controller. Enter 't' to host a Target
c
Initializing as Controller ......
Calling RTI_InitReq
Looking for pairing entry...........
Pairing entry found, 0x00
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
p
Calling RTI_PairReq
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
1
Sending 1
RTI_SendDataCnf called with status 0x00
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
2
Sending 2
RTI_SendDataCnf called with status 0x00
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
3
Sending 3
RTI_SendDataCnf called with status 0x00
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
z
Sending z
RTI_SendDataCnf called with status 0x00
Menu
'p' to pair
'0'-'9' to send ZRC command
'z' to send ZID keyboard 'z'
'q' to quit
```



```
C:\Users\         >C:\WORK\RemoTI_apps\RemoTI-CC253xDK-1.3                  \Proj
ects\tools\SimpleConsole\Debug\SimpleConsole.exe //./COM45
Support ZID? 'y' for yes
y
Enter 'c' to host a Controller. Enter 't' to host a Target
t
Initializing as Target ......
Calling RTI_InitReq
Looking for pairing entry...........
Pairing entry found, 0x00
Menu
'p' to allow pairing
'q' to quit
p
Calling RTI_AllowPairReq
Menu
'p' to allow pairing
'q' to quit
RTI_ReceiveDataInd called with len 1
Profile ID = ZID
[0x41]

Waiting for data from RC, press 'q' at any time to quit
RTI_ReceiveDataInd called with len 2
Profile ID = ZRC
 --user controlled pressed
[0x01,0x21]

RTI_ReceiveDataInd called with len 2
Profile ID = ZRC
 --user controlled pressed
[0x01,0x22]

RTI_ReceiveDataInd called with len 2
Profile ID = ZRC
 --user controlled pressed
[0x01,0x23]

RTI_ReceiveDataInd called with len 12
Profile ID = ZID
[0x02,0x0A,0x01,0x02,0x00,0x00,0x1D,0x00,0x00,0x00,0x00,0x00]

_
```

7. Conclusion

This application note has outlined the steps for correct project environment configuration using the RTI dll with Visual Studio C++ 2010 Express Edition. The sample code included with the application note also provides a very simple target node and remote controller node application. The uniform RTI application programmer interface across different platforms enables the Win32 environment for developing RemoTI application code and customer specific tools.

References

[1] RemoTI API, SWRA268

8. General Information

8.1 Document History

Revision	Date	Description/Changes
SWRA304	2009.09.03	Initial release.
SWRA304A	2009.09.09	Updated broken link to Microsoft VC++ 2008 Express Edition
SWRA304B	2012.04.09	Updated to match new release; RemoTI 1.3 and Microsoft VC++ 2010 Express Edition
SWRA304C	2013.01.18	Updated for RemoTI-1.3.1

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

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