Application Report
SPNA234–July 2016

Using the SPI as an Extra UART Transmitter

Shivani Iyer

ABSTRACT

It is quite common, especially when developing an application based on an entry level microcontroller, to prioritize peripheral usage based on the application's functional requirements. It may be the case that all of the available UARTs on a device are used for functional purposes, leaving no UARTs available for the developer to use for logging debug messages. This application report demonstrates an easy way to re-purpose a spare SPI port on a Hercules device for use as a UART transmitter so that debug messages can still be logged to a PC COM port even when all of the hardware UARTs on the device are being used for other purposes. Several features of the Hercules SPI port make this simple to implement with minimal software overhead. This application report illustrates the concept with a Hercules RM42 LaunchPad™ but can be adapted to any Hercules device with a spare SPI or MibSPI peripheral.

Project collateral and source code mentioned in this document can be downloaded from the following URL: http://www.ti.com/lit/zip/spnc049.

Contents

List of Figures

LaunchPad, Code Composer Studio are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

1 Introduction

It is quite common, especially when developing an application based on an entry level microcontroller, to prioritize peripheral usage based on the application's functional requirements. It may be the case that all of the available UARTs on a device are used for functional purposes, leaving no UARTs available for the developer to use for logging debug messages. This application report demonstrates an easy way to repurpose a spare SPI port on a Hercules device for use as a UART transmitter so that debug messages can still be logged to a PC COM port even when all of the hardware UARTs on the device are being used for other purposes. Several features of the Hercules SPI port make this simple to implement with minimal software overhead. This application report illustrates the concept with a Hercules RM42 LaunchPad but can be adapted to any Hercules device with a spare SPI or MibSPI peripheral.

The software included in this application report is a simple way to approach the problem mentioned above, and can easily be used with the RM42 LaunchPad. The code is written in C and requires only one SPI for operation. In the provided source example, the easily accessible SPI3 is used to transmit signals with a baud rate of 115,200. If SPI3 is being used for a different function, the necessary changes should be made to change the SPI being used.

2 Installation

2.1 Installing Software

The following software must be installed:

- HALCoGen: http://www.ti.com/tool/halcogen
- Code Composer Studio: http://processors.wiki.ti.com/index.php/Download_CCS
- SPNA234 Installer: http://www.ti.com/lit/zip/spnc049
- Terminal Program (for example, Tera Term): https://en.osdn.jp/projects/ttssh2/releases/

2.2 SPNA234 Installer Package

Click on the Installer Package to install the project contents.

1. Ensure that the installation location is as desired and click "Next".
2. Walk through all the installer steps, and once "Finish" is clicked, the appropriate folders will be downloaded to the specified location.
3. The default location is C:/ti/Hercules/appnotes/spiuart. Go to the folder and ensure that the "Sources" folder has been downloaded; this is the folder with the source code.

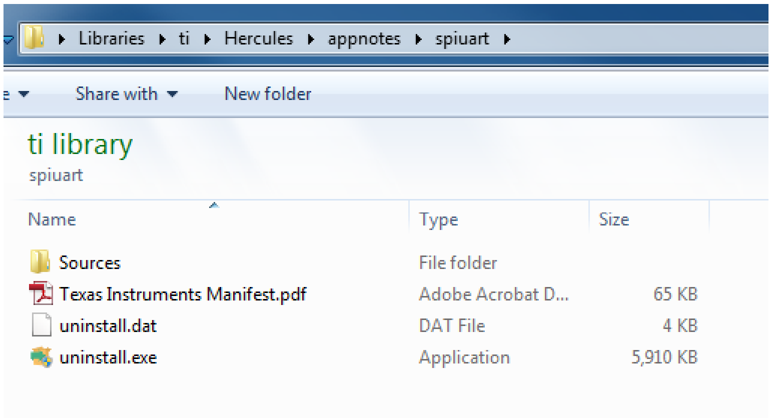

Figure 1. Sources Folder

2.3 Importing a Project

1. Open Code Composer Studio™ (CCS) and set a location for your workspace (any location is fine) and click the "OK" button.
2. To import the spi_uart project:
   (a) Navigate to File --> Import.
   (b) Click on C/C++ --> CCS Project.
   (c) Click the "Next" button.

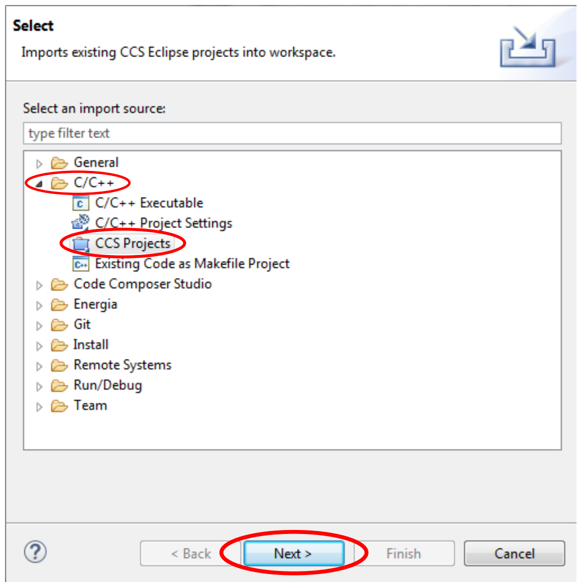

Figure 2. Import Project

3. Under "Select search-directory", click the "Browse" button and navigate to where the "Sources" file is located.
4. Select the file "Sources" and then click the "OK" button.

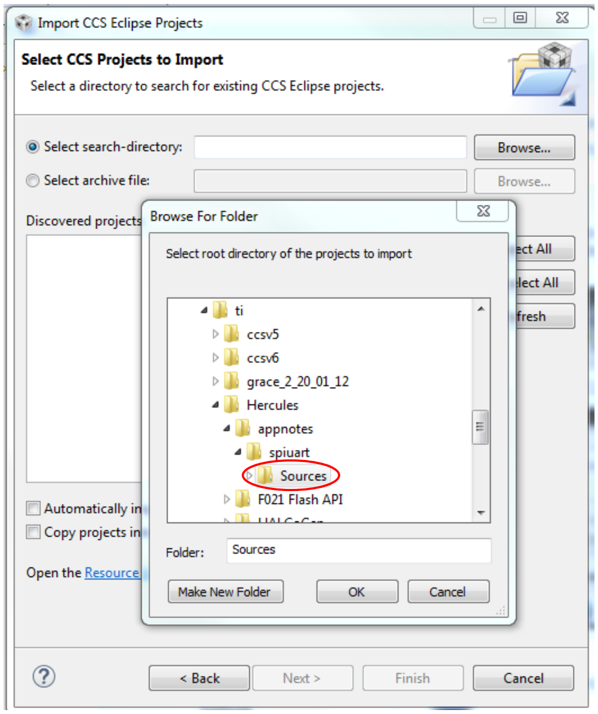

Figure 3. Select CCS Projects to Import

5. Click on the check box for "uart_spi" under the "Discovered projects" section and make sure "Automatically import referenced projects found in same search-directory" is checked off.
6. Click on the "Finish" button.

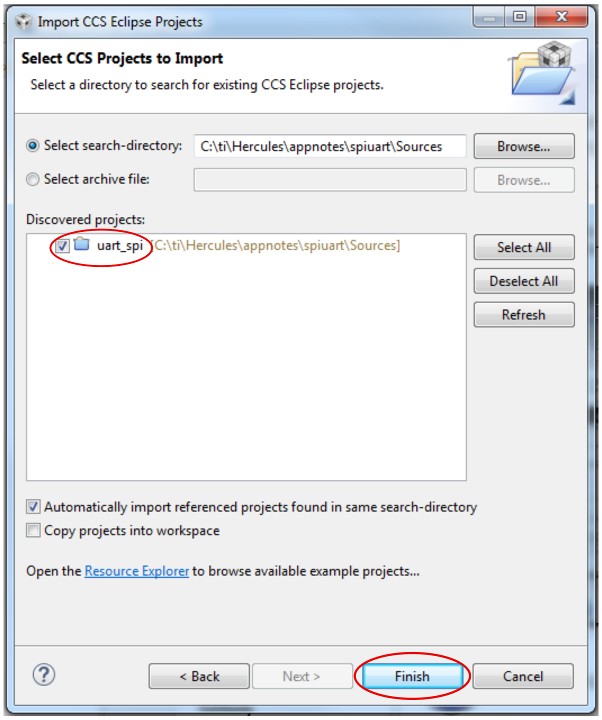

Figure 4. Discovered Projects

2.4 Include Options

The project will now show up in the Project Explorer toolbar on the left. If the toolbar is not visible, click on View --> Project Explorer.

1. Right-click on the project in the Project Explorer toolbar and click on "Properties".
   (a) Navigate to Build --> ARM Complier --> Include Options, and ensure that all the include options shown in Figure 5 are added.
2. Click the "OK" button.

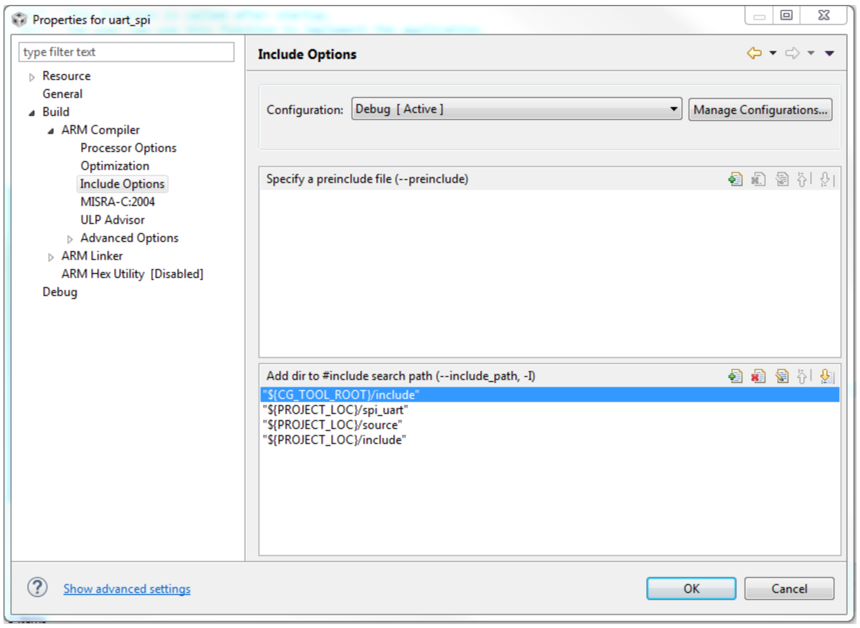

Figure 5. Include Options

3 Hardware

The hardware interface for this project includes:

- RM42 LaunchPad [1]
- RS-232 Level-Shifting Daughter Board [2]
- RS-232 cable

1. Place the Daughter Board onto the appropriate pins of the LaunchPad.
2. Connect the small end of the micro-USB cable to the LaunchPad and the larger end to a USB port on the computer.
3. Connect the DB9 connector of the RS-232 cable to the Daughter Board and the other end to the computer.
4. Connect a wire from the SPI3SIMO hole to the topmost UTX pin on the Daughter Board (this allows the data sent to the LaunchPad to travel back to the computer so it can be seen on the terminal window).

The setup of these two boards is shown in Figure 6.

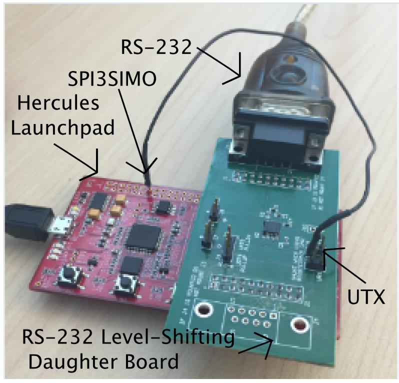

Figure 6. Hardware Setup

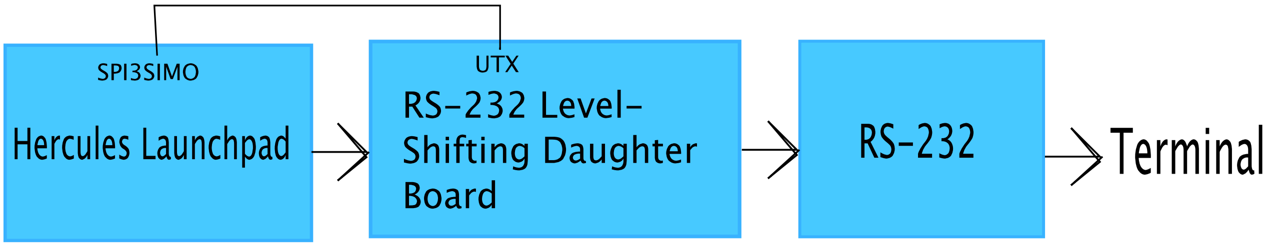

Figure 7. Hardware Block Diagram

4 Configurations

The configuration parameters are located in the header file *spi_uart.h*. The only configuration that the user has to be concerned about is the divider value in the prescale register for the baud rate, *EPRESCALE_FMT0*. A predetermined configuration is *START_STOP*, which is the value that adds the appropriate start and stop bits to the data. The default start bit is 0 and the default stop bit is 1. The SPI on this device makes it easy to configure the data sent out to be 10 bits long and shift LSB out first, which is the structure used in this application. The data is framed as shown in Figure 8.

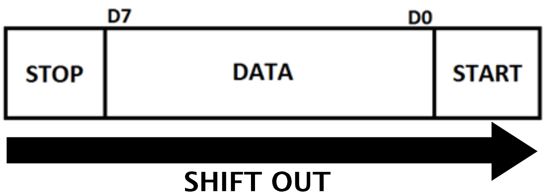

Figure 8. Data Framing

Figure 9 shows a sample configuration.

```
/*
 * Start bit, stop bit and baud rate configurations
 */
#define START_STOP 0xFE00
#define EPRESCALE_FMT0 0x363
```

Figure 9. Configurations

As seen above, the value of *EPRESCALE_FMT0* can be set to be any value to yield the desired baud rate for the UART. The formula to derive this is given in Figure 10. VBUSPCLK is the clock frequency.

$$BAUD\ RATE = \frac{VBUSPCLK}{EPRESCALE\_FMT0 + 1}$$

Figure 10. Formula to Derive Baud Rate

Using the value for *EPRESCALE_FMT0* from Figure 9, 0x363, and with a clock of 100 MHz, the resulting baud rate is approximately 115,207.

5 Demo

1. Open a terminal program (for example,Tera Term)
   (a) Set the port to "Serial" and choose the "USB Serial Port" option.
2. Click the "OK" button

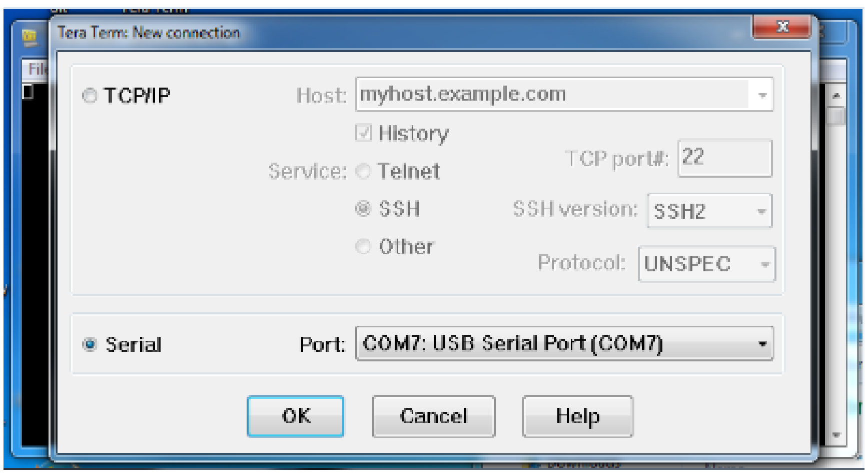

Figure 11. Tera Term Setup

3. If using Tera Term, navigate to Setup --> Serial Port... where the menu in Figure 12 pops up.
   (a) Change the baud rate to 115200 and leave the rest of the setup options unchanged
   (b) Click the "OK" button
4. If not using Tera Term, follow the appropriate steps to ensure:
   (a) There are 8 data bits
   (b) The baud rate is 115200 bits/s
   (c) There are no parity bits
   (d) There is 1 stop bit

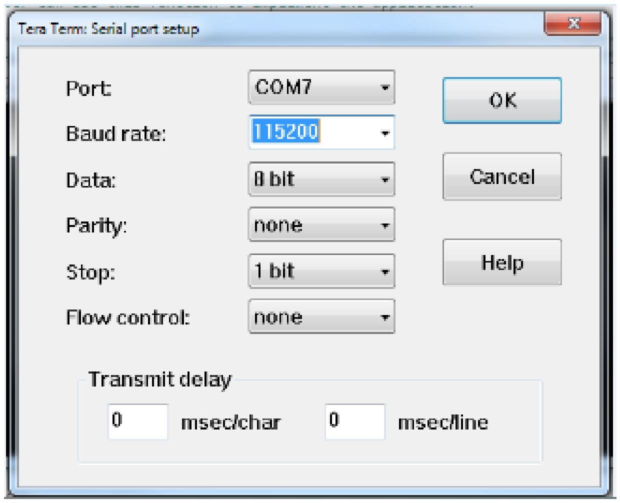

Figure 12. Tera Term Serial Port Setup

5. Go back to CCS and click on Project --> Build Project
6. If there are no errors, click on Run --> Debug
7. Once the program is ready to be run, click the resume button on the toolbar on the top shown in Figure 13

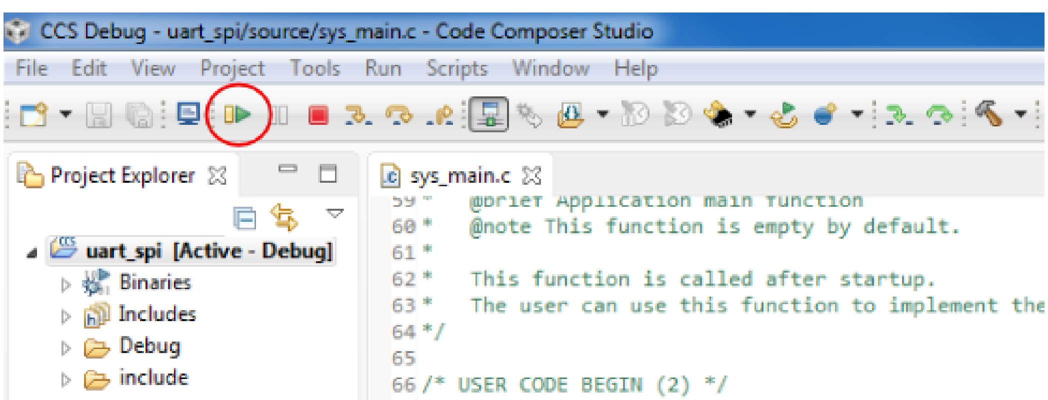

Figure 13. Resume

8. The following should show up on the terminal program once the resume button has been clicked.

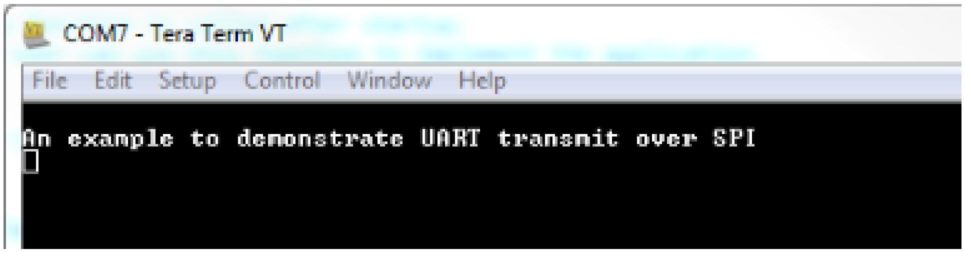

Figure 14. Tera Term Output

When using an oscilloscope, the width of each pulse can be seen to be close 115,200 bits per second and the start and stop bits are clearly visible, enclosing 8 bits of data between them. Figure 15 shows the oscilloscope capture when a string was sent. The data sent between the start and stop bits is 0x6C, which corresponds to the letter "l".

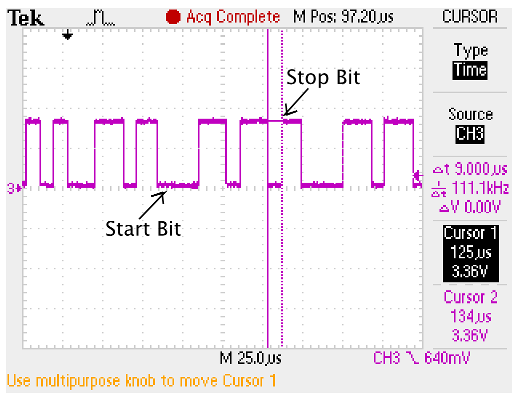

Figure 15. Oscilloscope Capture

6 References

1. RM42 LaunchPad: https://store.ti.com/LAUNCHXL-RM42.aspx
2. RS-232 Level-Shifting Daughter Board: http://www.ti.com/tool/TIDM-TM4C129XS2E

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**	
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 and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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