

Serial to Ethernet Converter for TM4C129x Microcontrollers Using RTOS

Sainandan Reddy

ABSTRACT

Some legacy products may only contain a serial port and accessing such products is increasingly becoming a challenge due to the inability to add multiple such products to a shared network and the inability to access the legacy product from long distances. A serial-to-Ethernet converter provides a simple solution to solve the above challenges. The software described in this application report allows an Ethernet-based TM4C129x microcontroller to be used as a serial-to-Ethernet converter. TM4C129x microcontroller is ideally suited for this application due to the integration of MAC and PHY on the microcontroller along with on-chip voltage regulator. This allows quicker time-to-market and cost savings. The software uses FreeRTOS to schedule the various tasks and runs out of the box on an EK-TM4C1294XL board.

Project collateral and source code discussed in this application report can be downloaded from the following URL: <http://www.ti.com/lit/zip/spma072>.

Contents

1	Introduction	2
2	Software Description	2
3	Set Up and Program <i>enet_s2e</i> Application.....	3
4	Demonstration.....	4
5	Configuration Webserver	6
6	Flow Control	11
7	Additional Hardware	11
8	References	14

List of Figures

1	Software Layers of <i>enet_s2e</i> Application.....	3
2	S2E Demonstration Representation.....	4
3	Finder Utility Displays the Boards Available on the Network	6
4	S2E Status Page.....	7
5	S2E Port 0 Settings Page	9
6	S2E Miscellaneous Settings Page	10
7	Schematic of RS-232 Level Shifter	12

List of Tables

1	Connections From RS-232 Level Shifter to TM4C129x Boards	13
2	Bill of Materials (BOM).....	13

Code Composer Studio, Tiva are trademarks of Texas Instruments.
 Keil is a registered trademark of ARM Limited.
 IAR is a trademark of IAR Systems AB.
 All other trademarks are the property of their respective owners.

1 Introduction

When a legacy product only contains a serial port for configuration or control, continuing to access the legacy product through the serial interface can become challenging over time. The following are the challenges of using a serial port:

- The inability to add multiple such products to a shared network
- The inability to access the legacy product from long distances
- Modern PCs generally do not have RS-232 ports

Using Ethernet in place of the serial port provides the following benefits:

- The legacy product can be more easily shared (instead of changing a cable connection, a new connection over the existing network is made)
- Ability to access from longer distances. Cable length is no longer an issue
- Ethernet is a more common interface available on today's computing equipment

The software described in this application report (*enet_s2e*) allows an Ethernet-enabled TM4C129x microcontroller to be used as a serial-to-Ethernet converter. By connecting a serial-to-Ethernet converter to the serial port of a legacy product, it can be given the ability to operate on the Ethernet without requiring any changes to the existing hardware or software. This is especially useful when the legacy product cannot be modified (such as in the case of third-party products).

The software works out of the box on an EK-TM4C1294XL board. With minor modifications, the software can run on the DK-TM4C129X board. For details, see [Section 3.2](#).

FreeRTOS has been used for scheduling the various tasks. Although an RTOS is not a requirement for this application, it has been used to make the application easily scalable.

2 Software Description

The main function of the *enet_s2e* application is to transfer information between serial peripherals and the Ethernet. To transfer information over the Ethernet, Telnet protocol is used. A number of features are provided in the application to make this process easy to use and robust. The following are some of these features.

- Two Serial-to-Telnet port pairs: Port0 and Port1
- Dynamic configuration of Serial and Telnet port settings
- Dynamic IP address configuration
- HTTP server to manage dynamic configuration parameters
- Locator service to find the IP address of the board (to access HTTP server)
- EEPROM access to store configuration data

The application runs a webserver to manage the dynamic/run-time configuration of many parameters. For more details of the configuration webserver and the different settings, see [Section 5.2](#).

This application uses two serial peripherals and two Telnet ports. The serial peripheral and Telnet port are configured such that each serial peripheral can transmit and receive data to and from only one Telnet port. There are two such serial-Telnet pairs, represented by Port0 and Port1. By default, the serial interface of Port0 and Port1 are assigned to UART4 and UART3 peripherals, respectively. This default peripheral assignment can be modified in *config.h* file. If the default peripheral assignment is modified, then the assignment of the interrupt handlers *SerialPort0IntHandler* and *SerialPort1IntHandler* in the vector table should also be modified. The vector table is located in the start-up file.

The Telnet ports can be configured in both server and client modes along with the necessary parameters like port number and IP address. By default, both the Telnet ports are configured as server. The Telnet ports used by Port0 and Port1 are 23 and 26, respectively.

Port0 and Port1 settings can be modified dynamically. You also have an option to make these modifications permanent. For more details about these settings, see [Section 5.2.2](#). The default values of these settings are stored in the global array *g_sParametersFactory* (present in the *config.c* file).

The *enet_s2e* application uses *FreeRTOS* to run multiple tasks in a concurrent fashion, a TCP/IP stack, called *lwIP*, to handle the TCP/IP packets and *TivaWare for C Series* that includes peripheral driver library and other utilities that simplify application development. Figure 1 shows the interaction between different software layers of this application.

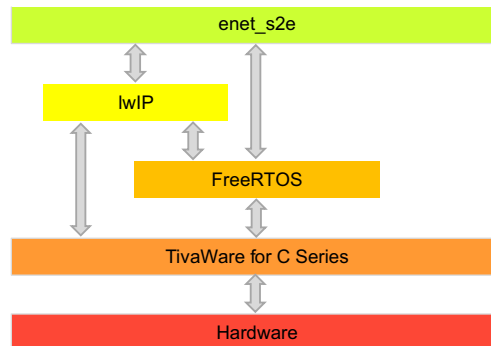


Figure 1. Software Layers of *enet_s2e* Application

The *enet_s2e.c* file contains the main function and performs the initialization before handing over control to the FreeRTOS scheduler. The tasks and their responsibilities are as follows:

- A TCP/IP task to run the lwIP stack and manage all the TCP/IP packets
- An Ethernet task to manage the Ethernet interface and its interrupt
- A serial task to manage the serial peripherals and their interrupts
- An idle task (automatically created by FreeRTOS) to manage changes to IP address

The TCP/IP task and Ethernet task are created and managed by the lwIP library.

The data transfer between the serial peripherals and telnet sockets is handled by the *TelnetHandler()* function (present in *telnet.c* file), which is called by *lwIPHostTimerHandler()* function (present in *lwip_task.c* file). The *lwIPHostTimerHandler()* function runs in the TCP/IP task context.

3 Set Up and Program *enet_s2e* Application

3.1 Set Up *enet_s2e* Application

1. Download and install *TivaWare for C Series* v2.1.0.12573 or later from the URL: <http://www.ti.com/tool/SW-TM4C>.
2. Download the *enet_s2e* application source code from the URL: <http://www.ti.com/lit/zip/spma072>. The zip file has two folders that need to be extracted: *enet_s2e* and *utils*.
3. Copy the *enet_s2e* folder to the EK-TM4C1294XL board's folder in *TivaWare for C Series*. The default installation path of *TivaWare for C Series* is *C:/ti* and the EK-TM4C1294XL board's folder is located at *../TivaWare_C_Series-2.1.0.12573/examples/boards/ek-tm4c1294xl/*.
4. The *utils* folder has two files: *eeprom_pb.c* and *eeprom_pb.h*. Copy them into the *utils* folder in *TivaWare for C Series* located in *../TivaWare_C_Series-2.1.0.12573/utils*.
5. Open the project in one of the toolchains, and build the project to create a binary file. The supported toolchains are Code Composer Studio™, Keil® MDK, IAR™ Embedded Workbench and GNU.

3.2 Software Modification for DK-TM4C129X

Use the following steps to run the *enet_s2e* application on the DK-TM4C129X board:

1. Download and extract the zip file as described in [Section 3.1](#). Then, copy the *enet_s2e* folder into the DK-TM4C129X board's folder located at `../TivaWare_C_Series-2.1.0.12573/examples/boards/dk-tm4c129x/`.
2. Open the project in one of the supported toolchains and modify the project settings as follows:
 - (a) Replace the labels `PART_TM4C1294NCPDT` and `TARGET_IS_TM4C129_RA1` with `PART_TM4C129XNCZAD` and `TARGET_IS_TM4C129_RA0` in the project settings.
 - (b) Uncomment the line that defines the label `DK_TM4C129X` in the *enet_s2e.c* file.
3. Rebuild the project.

3.3 Program the *enet_s2e* Application

1. Power up the microcontroller by connecting the micro (smaller) end of the USB cable to the USB connector on the EK-TM4C1294XL board. Then, connect the other end of the USB cable to a free USB port on the PC.
2. As soon as the EK-TM4C1294XL board is powered up, the ICDI and Stellaris Virtual Serial Port drivers are installed.
3. Program the binary to the microcontroller. Then, reset the board to run the *enet_s2e* application. Now, the EK-TM4C1294XL board starts behaving as an S2E module.

NOTE: To verify whether the ICDI drivers are installed correctly (or if there is trouble programming the binary), see the *Stellaris® In-Circuit Debug Interface (ICDI) and Virtual COM Port Driver Installation Instructions* ([SPMU287](#)).

4 Demonstration

A simple set up, like the one shown in [Figure 2](#), demonstrates the serial-to-Ethernet functionality of TM4C129x. [Section 4.1](#) and [Section 4.2](#) provide instructions on how to use two Terminal windows on a PC to loop data back and forth between a serial peripheral and Ethernet peripheral of the PC using the S2E module. Although either of the Serial-to-Telnet port pairs could be used for this demonstration, Port 0 is used in this documentation.

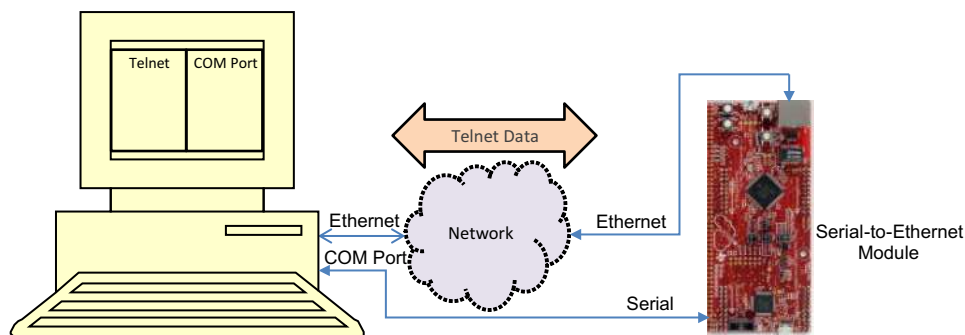


Figure 2. S2E Demonstration Representation

4.1 Set Up EK-TM4C1294XL Board and PC

1. Connect one end of the Ethernet cable to the Ethernet jack on EK-TM4C1294XL board and the other end to LAN with DHCP server present (like router).
2. For the schematic and connections for a simple circuit to level shift the CMOS/TTL signal of the universal asynchronous receiver/transmitter (UART) on EK-TM4C1294XL board to RS-232 levels, see [Section 7.1](#) and [Section 7.2](#).
3. Connect one end of a DB9 cable as described in [Section 7.1](#) and [Section 7.2](#) and the other end to the PC. If the PC does not have a DB9 port, then use a DB9 to USB converter cable.
4. Connect the micro (smaller) end of the USB cable to the USB connector on the EK-TM4C1294XL board. Then, connect the other end of the USB cable to a free USB port on the PC. This will power-up the board.

4.2 Demonstration

1. Open the first terminal window (like Hyperterminal or TeraTerm) and connect to the *Stellaris Virtual Serial Port* COM port. This port is used as a debug terminal to display the IP address and status messages. From now on, this terminal is referred to as debug terminal.
2. Select the Baud rate as 115200, Data Bits as 8, Parity as none, Stop bits as 1 and Flow Control as none.
3. Open a second terminal window and connect to the COM port associated with the port that the DB9 cable is connected to on the PC. This terminal will be referred to as Serial Terminal.
4. Select the Baud rate as 115200, Data Bits as 8, Parity as none, Stop bits as 1 and Flow Control as none.
5. Program and run the *enet_s2e* application. For detailed instructions to program the *enet_s2e* application, see [Section 3](#).
6. A banner and status messages are printed on the debug terminal. Proceed only after the IP address is acquired and displayed on the debug terminal.
7. Open a third terminal window to connect to the TCP/IP port 23 (or Telnet). This terminal will be referred to as Telnet terminal.
8. Select TCP/IP. If a particular protocol is required, choose Telnet. Set the Port number to 23. Set the Host IP address to the IP address shown on the debug terminal.
9. Information typed on either Serial or Telnet terminals will be displayed on the other terminal and vice versa. The text might not be displayed in the transmitting window based on the terminal windows setting.

This demonstrates that the S2E module routes back the serial data from PC as Telnet data (over Ethernet) and vice versa. In other words, the S2E module transmits Telnet data when it receives serial data and transmits serial data when it receives Telnet data.

The configuration webserver can be accessed by entering the IP address, displayed on the debug terminal, into a web browser. For more details on this webserver, see [Section 5](#).

5 Configuration Webserver

The *enet_s2e* application hosts an embedded webserver that provides a convenient configuration interface.

5.1 Access the Configuration Webserver

The configuration webserver of the S2E module can be accessed by:

- Using the Debug port
- Using the Finder utility

5.1.1 Using the Debug Port

To access the configuration webserver using the debug port:

1. Open a terminal window (like Hyperterminal or TeraTerm) and connect to the *Stellaris Virtual Serial Port* COM port.
2. Select the Baud rate as 115200, Data Bits as 8, Parity as none, Stop bits as 1 and Flow Control as none.
3. Program and run the *enet_s2e* application. For detailed instructions to program the *enet_s2e* application, see [Section 3](#).
4. As soon as the IP address is acquired, it is printed on the terminal.
5. Start a web browser and type the IP address that is displayed on the terminal into the address bar to load the different pages on the S2E configuration webserver.

5.1.2 Using the Finder Utility

To access the configuration webserver using the Finder utility:

1. Program and run the *enet_s2e* application. For detailed instructions to program the *enet_s2e* application, see [Section 3](#).
2. Browse to the `../TivaWare_C_Series-2.1.0.12573/tools/bin/` directory.
3. Double click on the *finder.exe* file. When the Finder utility starts, you should see the S2E module in the list of *Available Tiva Boards*, as shown in [Figure 3](#).
4. Start a web browser and type the IP address that is displayed in the Finder utility into the address bar to load the different pages on the S2E configuration webserver.

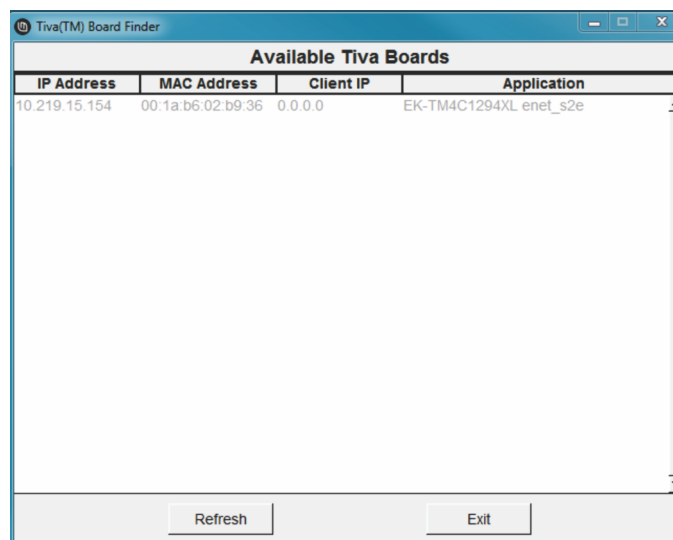


Figure 3. Finder Utility Displays the Boards Available on the Network

5.2 Different Pages Hosted on the Configuration Webserver

The configuration webserver hosts pages that give information about TI, Tiva™ C Series Microcontrollers, the TM4C1294NCPDT device and EK-TM4C1294XL board apart from the status and configuration pages that display the S2E status and configuration options. The status and configuration pages are covered in more detail in the following sections.

5.2.1 S2E Status Page

This page shows the current status and configuration information, as seen on [Figure 4](#). Included are the name, IP address, and MAC address of the S2E module as well as the current port settings for both port 0 and port 1.

S2E Status

Name: **TM4C1294XL S2E Example Application**
 IP Address: 10.219.15.154
 MAC Address: 00-1a-b6-02-b9-36

Current port settings:

	Port 0	Port 1
Baud Rate:	115200 bits/second	115200 bits/second
Data Size:	8 bits/character	8 bits/character
Parity:	None	None
Stop Bits:	1 bit(s)	1 bit(s)
Flow Control:	None	None
Local Telnet Port Number:	23	26
Remote Telnet Port Number:	N/A	N/A
Telnet Mode:	Server	Server
Telnet Protocol:	Telnet	Telnet
Telnet Server IP:	N/A	N/A
Telnet Timeout:	0 seconds	0 seconds

Copyright © 2015 Texas Instruments Incorporated. All rights reserved.

Figure 4. S2E Status Page

5.2.2 S2E Port 0 and Port 1 Settings Page

The “Port 0 Settings” page allows configuration of port 0 of the S2E module. Similarly, the “Port 1 Settings” page allows configuration of port 1 of the S2E module. The following configuration options are provided:

- Baud Rate – specifies the baud rate to be used by the serial port. There are several options up to 230400 bits/S.
- Data Size – configures the number of data bits per character. The options are 5 to 8 bits per character.
- Parity – specifies the generation and checking of the parity bit in the data frame and the type of parity used. The options are “None”, “Odd”, “Even”, “Mark”, and “Space”.
- Stop Bits – specifies the number of stop bits at the end of a frame. The options are 1 and 2 bits per character.
- Flow Control – specifies the use of flow control. The options are “None” and “Hardware”.
- Local Telnet Port Number – specifies the local telnet port number to be used.
- Remote Telnet Port Number – specifies the remote telnet port number to be used when the “Telnet Mode” is set to “Client”.
- Telnet Mode – specifies whether the telnet mode for that port will be “Server” or “Client”.
- Telnet Protocol – specifies whether the data for the port will be “Telnet” or “Raw”.
- Telnet Server IP – specifies the IP address of the telnet server when the “Telnet Mode” is set to “Client”.
- Telnet Timeout – specifies the telnet timeout in seconds. The default is 0 and specifies that no timeout is to be used.

After changing the settings, click the “Submit” button. If the “Make these the default settings” check box is checked before clicking the “Submit” button, then the new settings are applied each time the S2E module is reset. Otherwise, the existing defaults are used whenever the module is next reset. [Figure 5](#) shows the Port 0 Settings page.

The screenshot shows a web browser window at 10.219.15.154/s2e.shtm. The page title is "Tiva™ C Series TM4C1294XL Evaluation Kit EK-TM4C1294XL". The left sidebar contains a navigation menu with items like "About TI", "Tiva™ C Series Overview", "TM4C1294NCPDT Block Diagram", "EK-TM4C1294XL Product Page", "Tiva™ TM4C129x Family Product Page", "S2E Status", "S2E Port 0 Settings", "S2E Port 1 Settings", and "S2E Miscellaneous Settings". The main content area is titled "Port 0 Settings" and displays the following information:

Name: TM4C129XL S2E Example Application
IP Address: 10.219.15.154
MAC Address: 00-1a-b6-02-b9-36

Settings:
 The current settings for port 0 may be changed using the form below. To make the new settings apply each time the S2E module is reset, ensure that "Make these the default settings" is checked before pressing the "Apply Changes" button. If this control is not checked, the changes are applied to the port but the existing defaults are used whenever the module is next reset.

	Current	Updated
Baud Rate:	115200bits/second	230400 bits/S
Data Size:	8bits/character	5 bits/character
Parity:	None	None
Stop Bits:	1bit(s)	1 bit(s)
Flow Control:	None	None
Local Telnet Port Number:	23	
Remote Telnet Port Number:	N/A	
Telnet Mode:	Server	Server
Telnet Protocol:	Telnet	Telnet
Telnet Server IP:	N/A	
Telnet Timeout:	0seconds	seconds (< 256, 0 for no timeout)

At the bottom of the settings table, there is a "Submit" button and a checkbox labeled "Make these the default settings." The footer of the page reads "Copyright © 2015 Texas Instruments Incorporated. All rights reserved."

Figure 5. S2E Port 0 Settings Page

5.2.3 S2E Miscellaneous Settings Page

The Miscellaneous Settings page, shown in [Figure 6](#), is divided into two parts: “IP Address Selection” and “Restore Factory Defaults”.

The *IP Address Selection* portion of the page allows configuration of the S2E module to automatically obtain an IP address or use a static IP address at start up. If the “DHCP/AutoIP” option is chosen, the S2E module will first attempt to get an IP address from a DHCP server. If a DHCP server cannot be located, the S2E module will obtain a link local IP address using the AutoIP protocol. If the “Static IP” option is chosen, the “Static IP Address”, “Subnet Mask”, and “Default Gateway” fields need to be filled in. Clicking the “Update Settings” button causes the settings to be saved.

The *Restore Factory Defaults* portion of the page allows restoring all of the options to their factory default states.

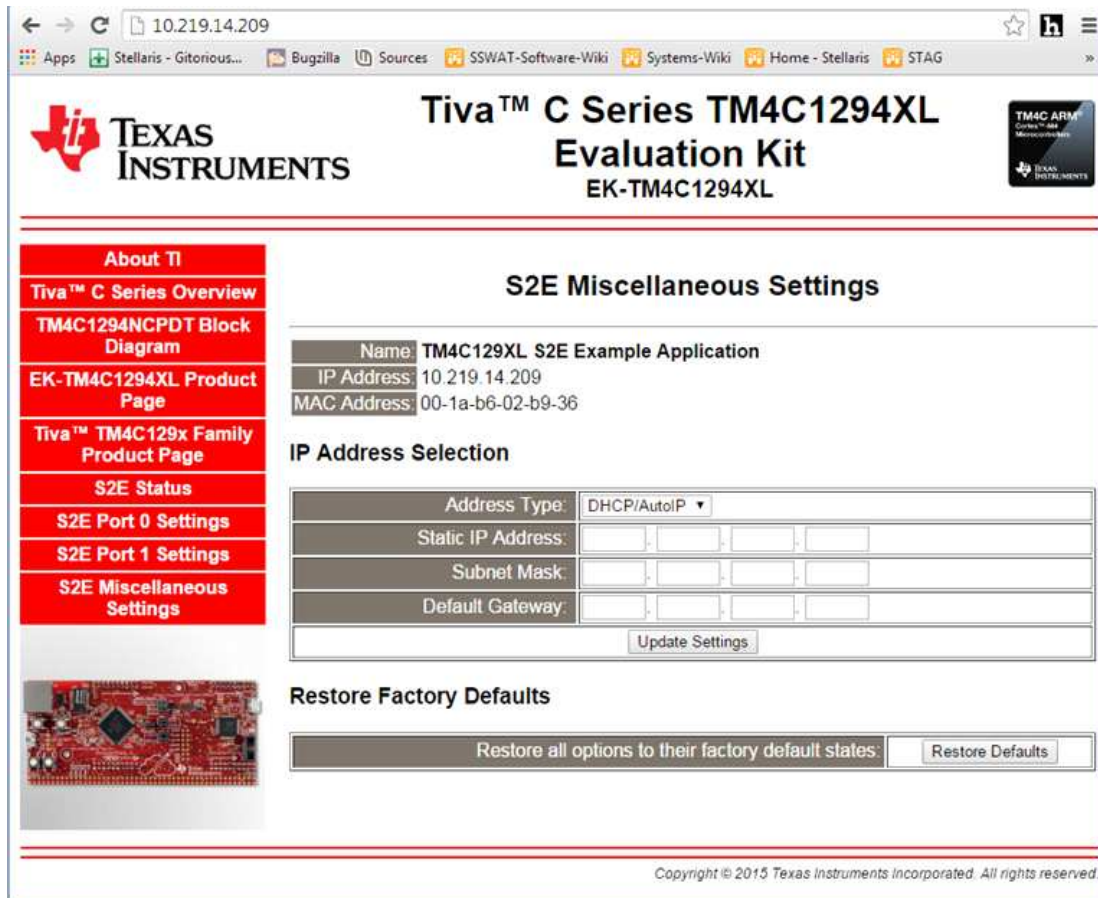


Figure 6. S2E Miscellaneous Settings Page

6 Flow Control

The *enet_s2e* application can be configured to use the RTS/CTS hardware flow control using the configuration webserver.

While receiving data in Flow Control mode, the S2E module asserts the RTS signal when it is capable of receiving data and it de-asserts the signal when transmission by the remote serial device should be paused. In the S2E module, the RTS signal is asserted and de-asserted by the UART peripheral.

While transmitting data, the CTS signal is monitored by the UART peripheral for state changes. When the state changes to asserted, transmission is allowed and resumed. When the state changes to de-asserted, transmission is paused.

The polarity of the CTS and RTS signal is interpreted in the same way. If the CTS signal is 1 (or high level), it is considered to be de-asserted, and the UART transmitter is disabled (the current byte, if any, is completed). When the signal is 0 or low level, it is considered to be asserted, and the transmitter is re-enabled.

7 Additional Hardware

For the demonstration described in [Section 4](#), the UART on the TM4C129x microcontroller has to communicate with the PC over COM port. To achieve this, the UART signals should be level shifted to RS-232 level. The EK-TM4C1294XL and DK-TM4C129XL boards don't have a level shifter. A schematic for level shifting is shown in [Section 7.1](#). Only one UART port with hardware flow control can be used with this schematic. [Section 7.2](#) lists the jumper settings and the necessary connections from EK-TM4C1294XL or DK-TM4C129XL.

7.1 Schematic of RS-232 Level Shifter

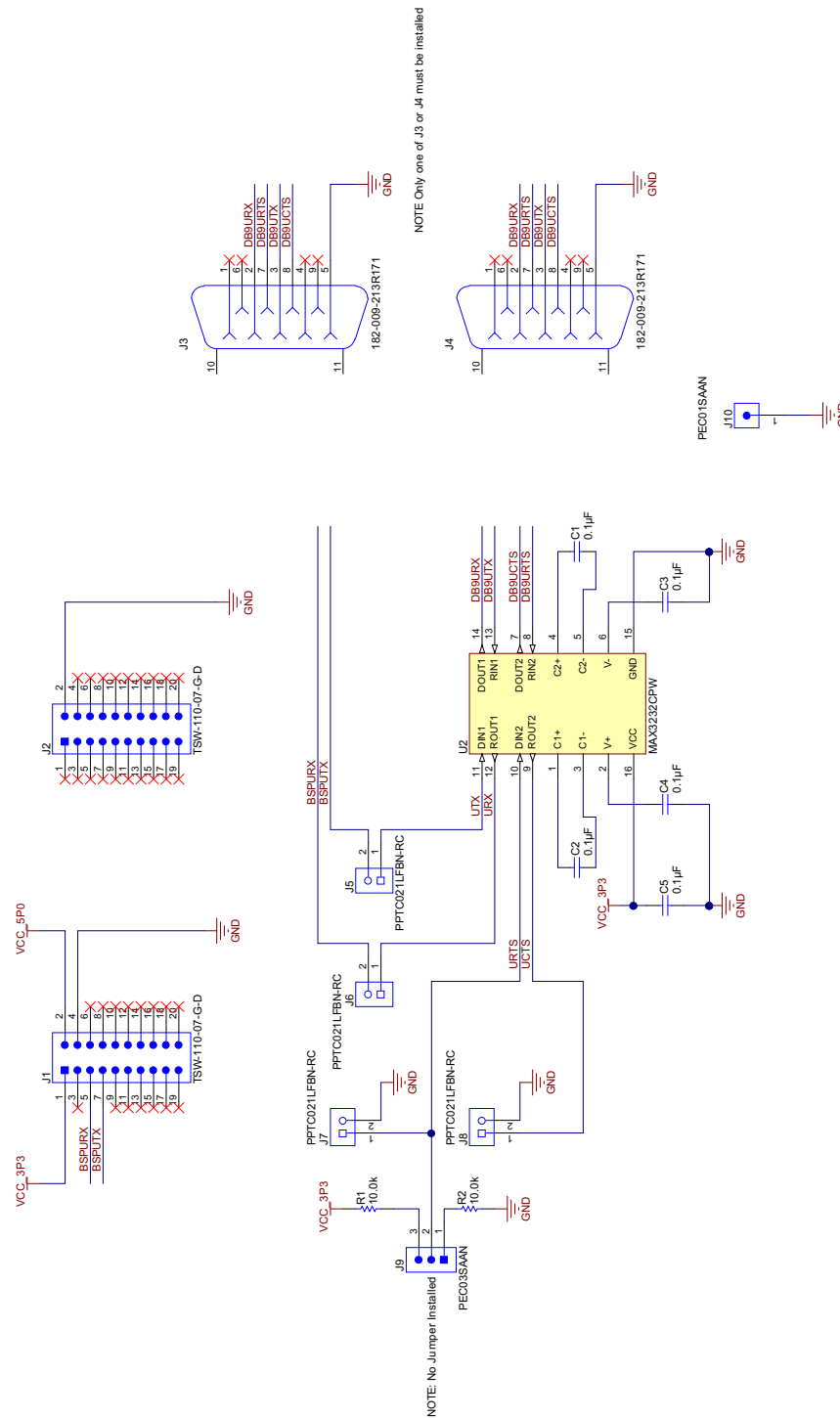


Figure 7. Schematic of RS-232 Level Shifter

7.2 Connections

The following connections are needed to demonstrate the *enet_s2e* application.

- For 3.3 V VCC and GND, connect J1.1 and J1.4 on the schematic shown on [Figure 7](#) to the following items, respectively:
 - X6.1 and X6.4 on the EK-TM4C1294XL board
 - J29.1 and J29.4 on the DK-TM4C129X board
- Follow the connection in [Table 1](#) to level shift the UART signals to RS232 level
- Connect J3 (or J4) on the schematic shown in [Figure 7](#) to a male-male DB9 cable. The other end of the DB9 cable goes to the PC.

[Table 1](#) lists the connections from the schematic depicted in [Figure 7](#) to either the EK-TM4C1294XL or DK-TM4C129X boards to achieve RS232 level shifting of the UART signals. The UART signal name is also mentioned in this table along with the Serial-To-Telnet port number that is used by the *enet_s2e* application.

Table 1. Connections From RS-232 Level Shifter to TM4C129x Boards

<i>enet_s2e</i> Port Number	UART Signal Name - GPIO	RS-232 Level Shifter Schematic	EK-TM4C1294XL	DK-TM4C129X
Port 0	U4RX – PK0	URX – J6.1	X6.10	J29.10
	U4TX – PK1	UTX – J5.1	X6.12	J29.12
	U4RTS – PK2	URTS – J7.1	X6.14	J29.14
	U4CTS – PK3	UCTS – J8.1	X6.16	J29.16
Port 1 ⁽¹⁾	U3RX – PA4	URX – J6.1	X6.18	J9.6 ⁽²⁾
	U3TX – PA5	UTX – J5.1	X6.20	J9.7 ⁽²⁾
	U3RTS – PN4	URTS – J7.1	X6.19	J27.16
	U3CTS – PN5	UCTS – J8.1	X6.17	J36.1 ⁽³⁾

⁽¹⁾ Another board has to be used for Port 1 connections.

⁽²⁾ Pins 2, 3 on J16 and J17 should be connected.

⁽³⁾ Remove the jumper on J36.

7.3 Bill of Materials (BOM)

Table 2. Bill of Materials (BOM)

Digi-Key Part Number	Manufacturer	Part Number	Designator	Qty	Supplier 1	Supplier Part Number 1
445-1266-1-ND	TDK	C1005X5R0J104K	C1, C2, C3, C4, C5	5	Digi-Key	445-1266-1-ND
SAM1028-10-ND	Samtec	TSW-110-07-G-D	J1, J2	2	Digi-Key	SAM1028-10-ND
182-909FE-ND	NorComp	182-009-213R171	J3, J4	2	Digi-Key	182-909FE-ND
S7000-ND	Sullins Connector Solutions	PPTC021LFBN-RC	J5, J6, J7, J8	4	Digi-Key	S7000-ND
S1012E-03-ND	Sullins Connector Solutions	PEC03SAAN	J9	1	Digi-Key	S1012E-03-ND
S1012E-01-ND	Sullins Connector Solutions	PEC01SAAN	J10	1	Digi-Key	S1012E-01-ND
P10.0KLCT-ND	Panasonic	ERJ-2RKF1002X	R1, R2	2	Digi-Key	P10.0KLCT-ND
296-26153-5-ND	Texas Instruments	MAX3232CPW	U1	1	Digi-Key	296-26153-5-ND

8 References

- TivaWare™ for C Series: <http://www.ti.com/tool/SW-TM4C>
- *Stellaris® In-Circuit Debug Interface (ICDI) and Virtual COM Port Driver Installation Instructions (SPMU287)*
- FreeRTOS: <http://www.freertos.org/>
- lwIP: <http://lwip.wikia.com/>

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

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com