

TI Designs

USB High Speed Reference Design for ARM® Cortex®-M4F Based High-Performance TM4C129x MCUs



Description

This TI Design uses Texas Instrument's high-performance ARM® Cortex®-M4F based TM4C129x microcontrollers (MCUs) with an integrated USB 2.0 controller to interface with external high speed USB PHY. The design includes software that allows data exchange between USB high speed and Ethernet. The design also features transceivers for CAN and UART serial interfaces to connect a host to legacy RS232 machines or to interface with controller area network (CAN) bus. Provision for digital communication interfaces like UART, I²C, and SSI to bridge external devices and aggregate data from slow interfaces to a high-speed USB link has been provided in this reference design.

Resources

TIDM-TM4C129USBHS	Design Folder
TPS737	Product Folder
SN65HVD1050	Product Folder
MAX3232	Product Folder
TPD2E2U06	Product Folder
TS3USB221E	Product Folder
TXB0106	Product Folder

Features

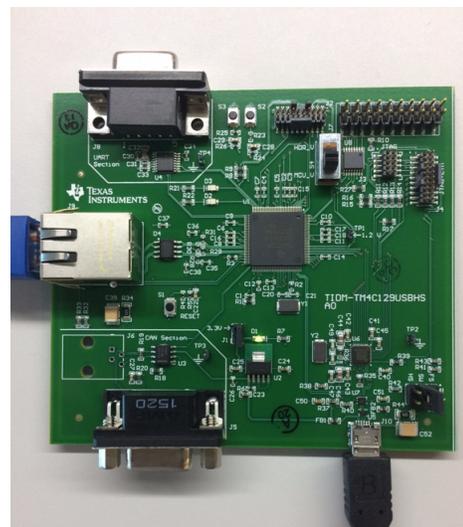
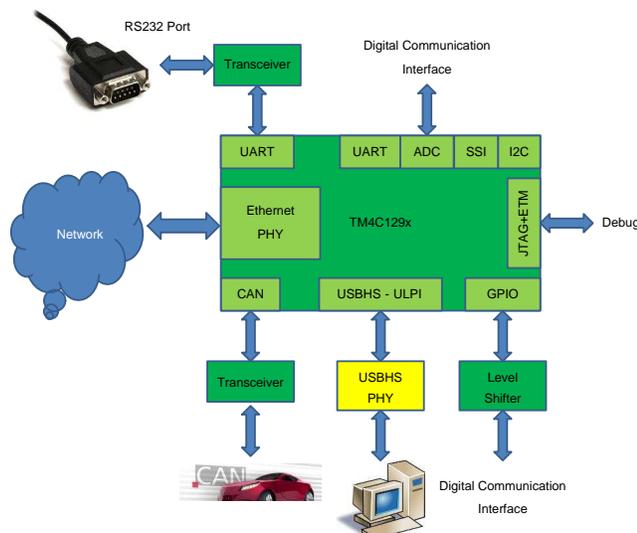
- USB 2.0 High-Speed PHY integration With Software-Controlled Switch to Enumerate USB Port as Either High Speed or Full Speed
- TivaWare™ Software USB Library for Ease of Developing Custom USB Software
- Firmware Example and Host Application Demonstrating Data Transfer from USB to Ethernet.
 - Example Code May Extend to Support Data Transfer From UART, CAN, SPI, and I²C to USB
- PCB Supports Connection to External RS232, CAN, and Ethernet Networks

Applications

- Industrial Communication Switch
- Programmable Logic Controller
- PLC Communication Module
- Instrumentation and Data Acquisition
- Single Board Computers



[ASK Our E2E Experts](#)





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

1 System Overview

1.1 System Description

The TM4C129x family of MCUs from Texas Instruments features a USB 2.0 high-speed OTG controller with ULPI interface for data transfers up to 480 Mbps. A high-speed USB link enables the TM4C129x MCU to transfer large amounts of data to and from a standalone host to the MCU. The data can then be exchanged efficiently over Ethernet. A high-speed USB link may be used for aggregating data from low-speed data interfaces like CAN, RS232, I²C, and SSI then sending the aggregated data to control applications running on a PC host. With an onboard transceiver for CAN and RS232, this design makes application development easier for the customer. The design files for this reference design include schematics, bill of materials (BOM), layer plot, Altium files, Gerber files, and embedded firmware.

1.2 Block Diagram

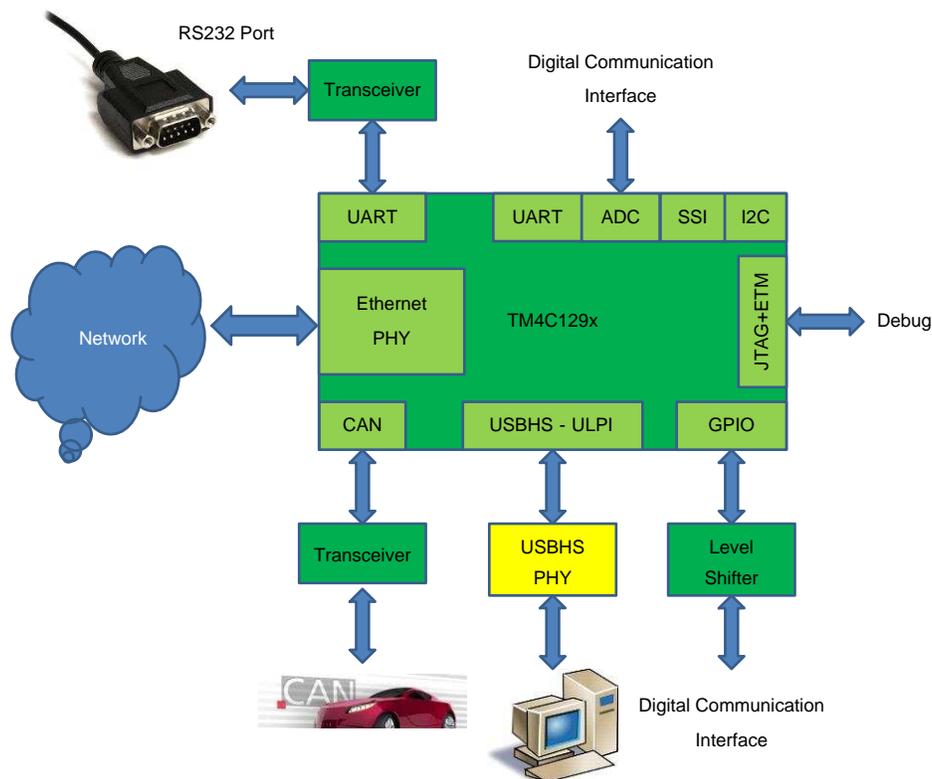


Figure 1. TIDM-TM4C129USBHS Block Diagram

1.3 Highlighted Products

1.3.1 TM4C129ENCPDT

The TM4C129ENCPDT is a 120-MHz high-performance MCU with 1-MB on-chip flash and 256-KB on-chip SRAM. The device features an integrated Ethernet MAC+PHY for connected applications and cryptographic modules of AES, DES, and SHA for encryption, decryption, and authentication. The device has high-bandwidth interfaces like a memory controller and a high-speed USB 2.0 digital interface. With integration of a number of serial communication peripherals, a 12-bit ADC capable of up to 4 MSPS, and motion control peripherals, the device provides a unique solution for a variety of applications ranging from industrial communication equipment to smart energy and smart grid applications.

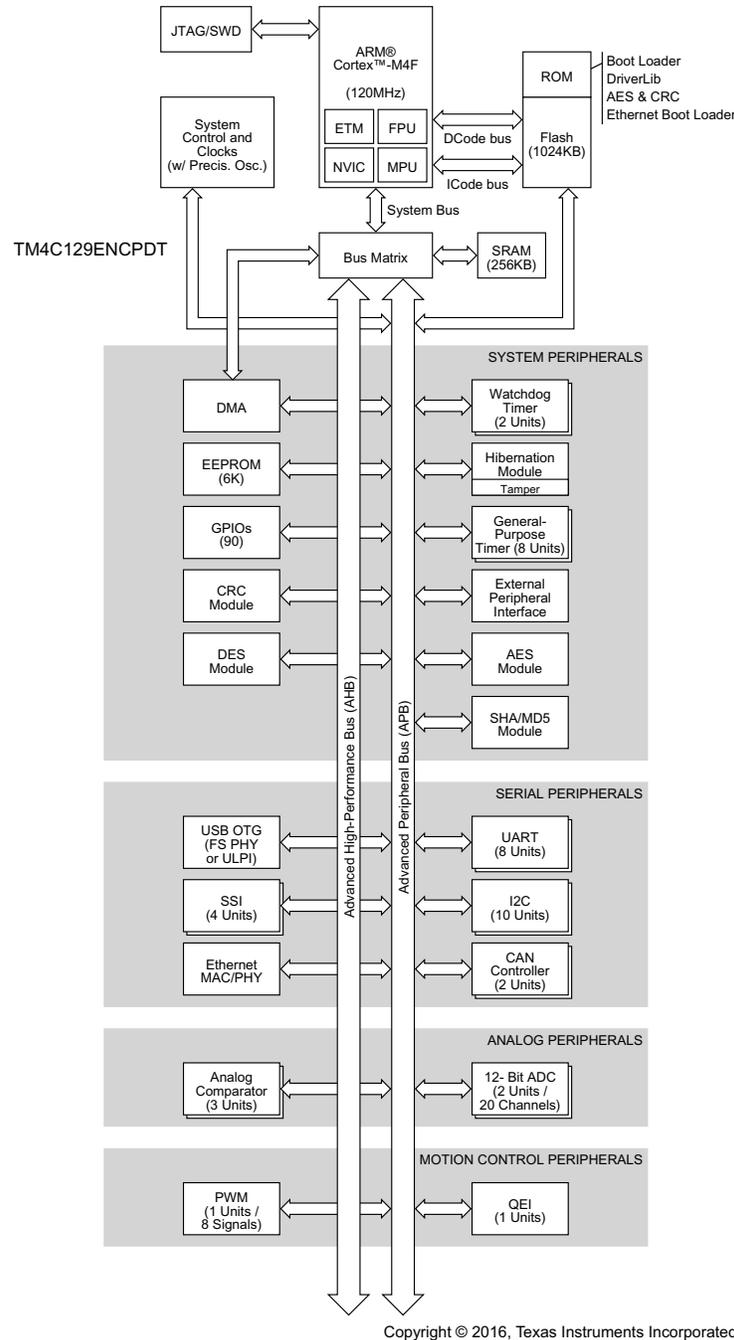


Figure 2. TM4C129ENCPDT MCU High-Level Block Diagram

1.3.2 MAX3232

The MAX3232 device consists of two line drivers, two line receivers, and a dual charge-pump circuit with ± 15 -kV electrostatic discharge (ESD) protection terminal to terminal (serial-port connection terminals, including GND). The device meets the requirements of TIA/EIA-232-F and provides the electrical interface between an asynchronous communication controller and the serial-port connector. The charge pump and four small external capacitors allow operation from a single 3- to 5.5-V supply. The devices operate at data signaling rates up to 250 kbps and a maximum of 30-V/ μ s driver output slew rate

1.3.3 SN65HVD1050

The SN65HVD1050 meets or exceeds the specifications of the ISO 11898 standard for use in applications employing a CAN. As a CAN transceiver, this device provides differential transmit capability to the bus and differential receive capability to a CAN controller at signaling rates up to 1 Mbps. The SN65HVD1050 is designed for operation in especially harsh environments. As a result, the device features cross-wire, overvoltage, and loss of ground protection from -27 V to 40 V, over-temperature shutdown, a -12 -V to 12-V common-mode range, and will withstand voltage transients from -200 V to 200 V according to ISO 7637. A dominant time-out circuit in the SN65HVD1050 prevents the driver from blocking network communication with a hardware or software failure. The time-out circuit is triggered by a falling edge on TXD (pin 1). If no rising edge is seen before the time-out constant of the circuit expires, the driver is disabled. The circuit is then reset by the next rising edge on TXD.

1.3.4 TPD2E2U06

The TPD2E2U06 is a dual-channel, low-capacitance TVS diode ESD protection device. The device offers ± 25 -kV contact and ± 30 -kV air-gap ESD protection in accordance with the IEC 61000-4-2 standard. The 1.5-pF line capacitance of the TPD2E2U06 makes the device suitable for a wide range of applications.

1.3.5 TS3USB221E

The TS3USB221E is a high-bandwidth switch specially designed for the switching of high-speed USB 2.0 signals in handset and consumer applications, such as cell phones, digital cameras, and notebooks with hubs or controllers with limited USB I/Os. The wide bandwidth (1 GHz) of this switch allows signals to pass with minimum edge and phase distortion. The device multiplexes differential outputs from a USB host device to one of two corresponding outputs. The switch is bidirectional and offers little or no attenuation of the high-speed signals at the outputs. The TS3USB221E is designed for low bit-to-bit skew and high channel-to-channel noise isolation and is compatible with various standards, such as high-speed USB 2.0 (480 Mbps).

1.3.6 TXB0106

This 6-bit non-inverting translator uses two separate configurable power-supply rails. The A port is designed to track VCCA. VCCA accepts any supply voltage from 1.2 V to 3.6 V. The B port is designed to track VCCB. VCCB accepts any supply voltage from 1.65 V to 5.5 V. This allows for universal low-voltage, bidirectional translation between any of the 1.2 V, 1.5 V, 1.8 V, 2.5 V, 3.3 V, and 5 V voltage nodes. VCCA should not exceed VCCB. When the output-enable (OE) input is low, all outputs are placed in the high-impedance state. The TXB0106 is designed so that the OE input circuit is supplied by VCCA. This device is fully specified for partial-power-down applications using Ioff. The Ioff circuitry disables the outputs, which prevents damaging current backflow through the device when it is powered down.

2 Getting Started Hardware and Software

This section describes the layout of the board for the TI Design and the steps to run the software example for connecting to Ethernet through the USB HS connection to a PC.

2.1 Board Layout Description

2.1.1 High Speed Interfaces

The board's high-speed interfaces consist of USB connector (J9) and 10/100 Mbps Ethernet connector (J10). [Figure 3](#) shows the position of the high-speed interfaces on the board.

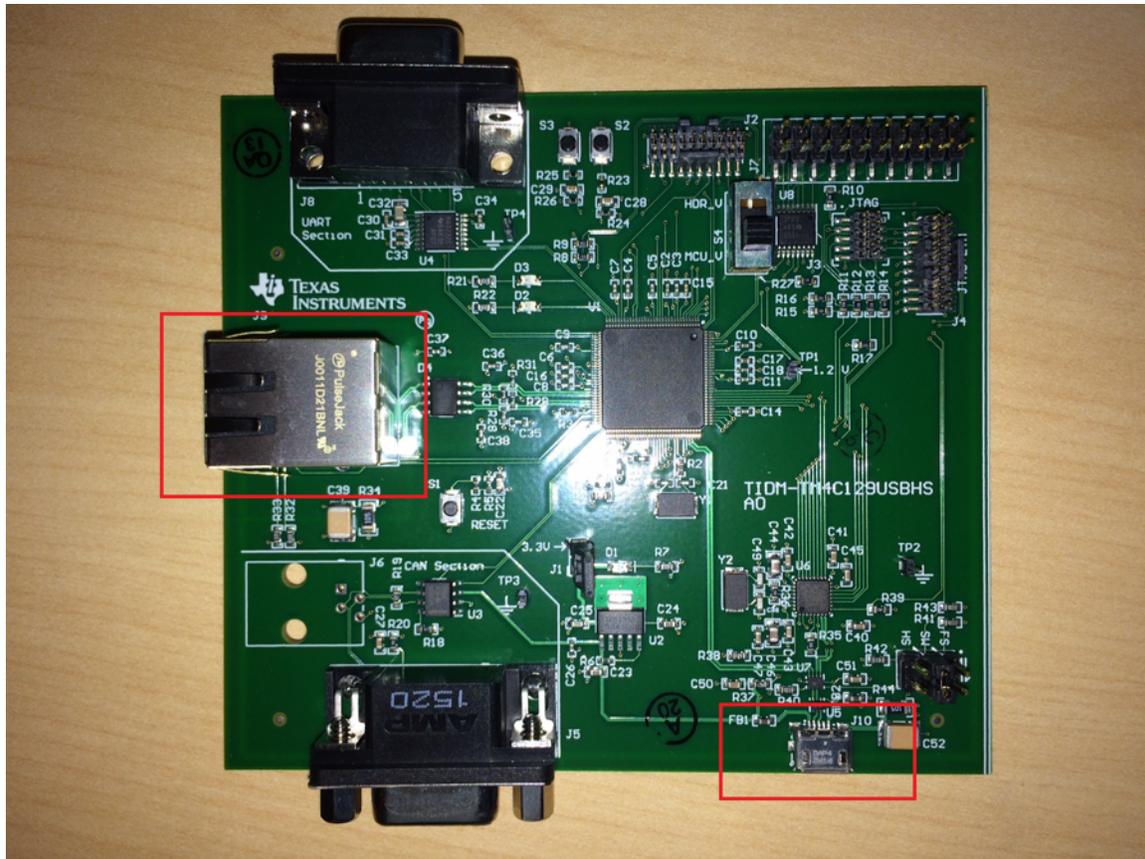


Figure 3. High Speed Interfaces

The USB connector has a set of jumper (J11) that is unpopulated and allows the user to configure the USB switch control (see [Figure 4](#)). [Table 1](#) shows the options that the user may use to configure the USB.

Table 1. USB Jumper Selection

JUMPER POSITION	COMMENTS
FS	USB connector connected to TM4C129x integrated USBFS PHY
SW	Connector selection controlled by GPIO PD7. When GPIO PD7 is high, the connector is connected to integrated USBFS PHY. When GPIO PD7 is low, the connector is connected to onboard USBHS PHY.
HS	USB connector connected to onboard USBHS PHY

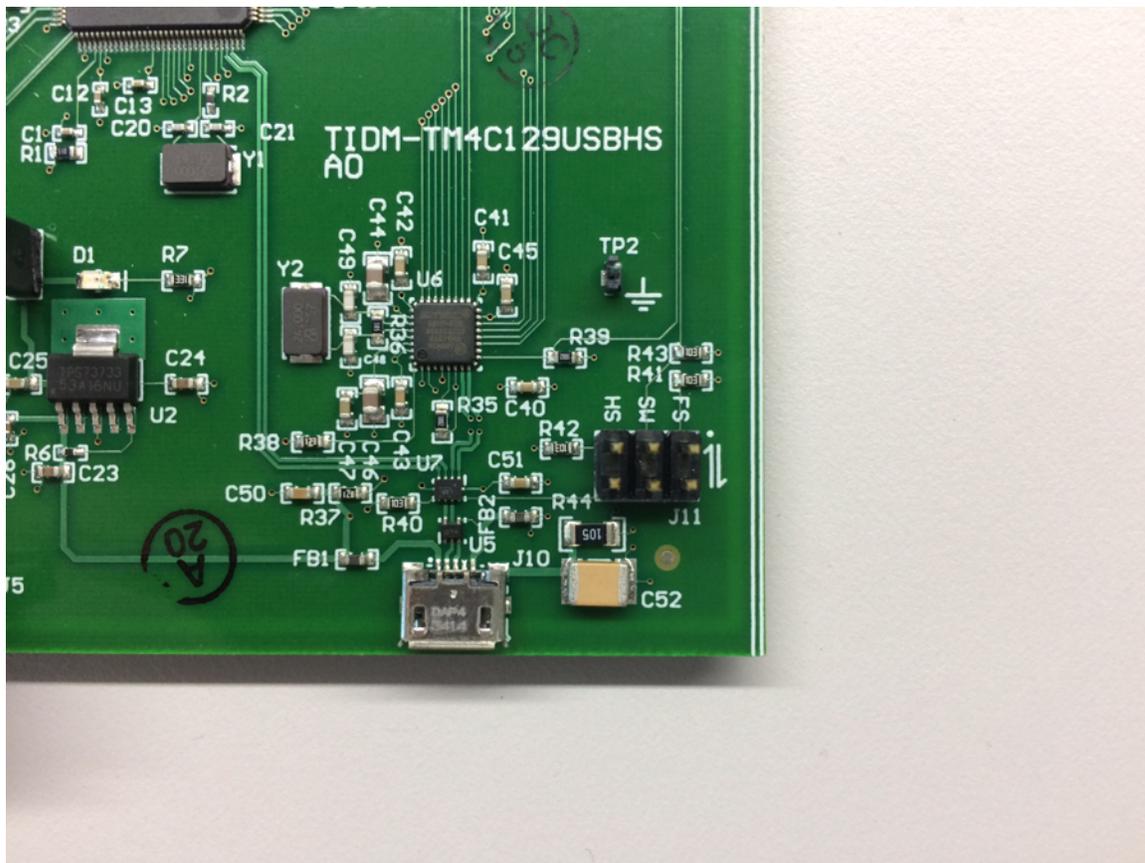


Figure 4. USB Jumper Position

2.1.2 Low-Speed Transceiver Interfaces

The board's low-speed interfaces consist of CAN connectors (J5 and J6) and RS232 connector (J8) via on board transceivers. [Figure 5](#) shows the position of the low-speed transceiver interfaces on the board. Note that connector J6 is unpopulated on the board as connector J5 is populated.

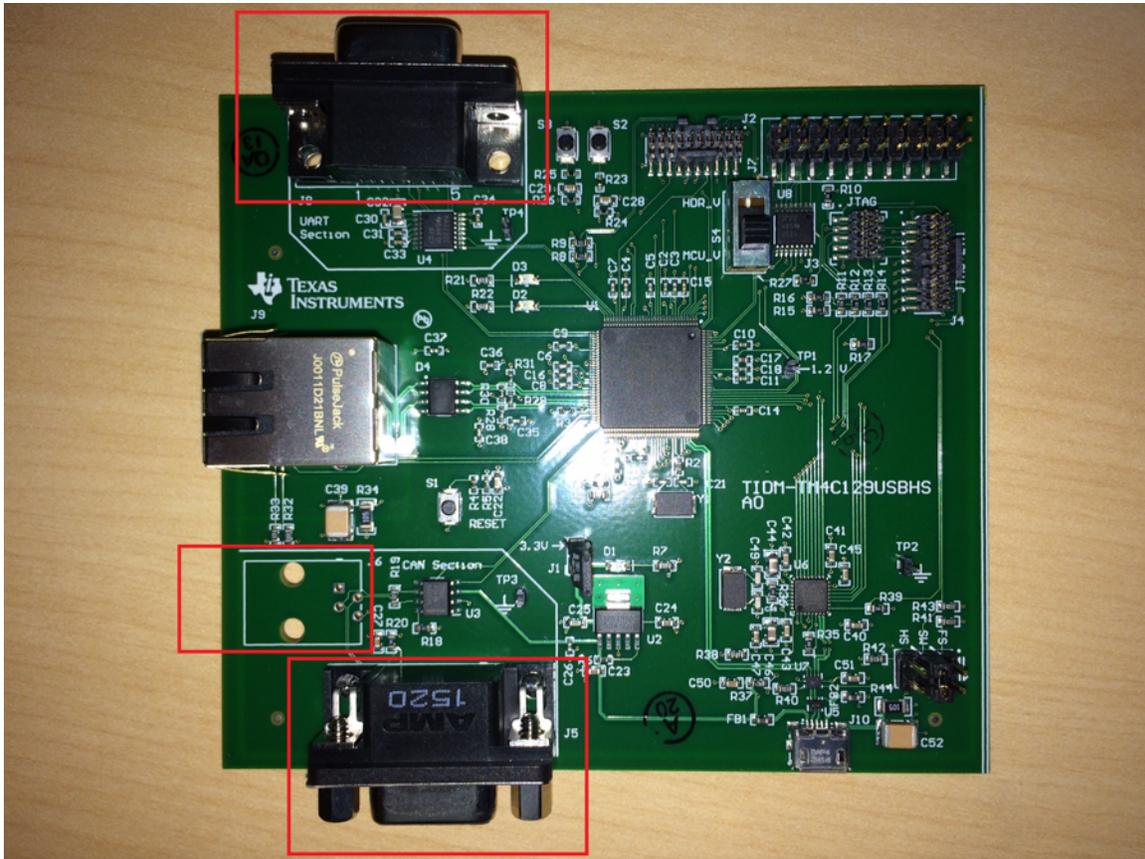


Figure 5. Low-Speed Transceiver Interfaces

2.1.3 Low-Speed Interfaces

The board's low-speed interfaces consist of J2 and J7 header, which map out GPIO, I²C, SSI, UART, and ADC channels (see [Figure 6](#)).

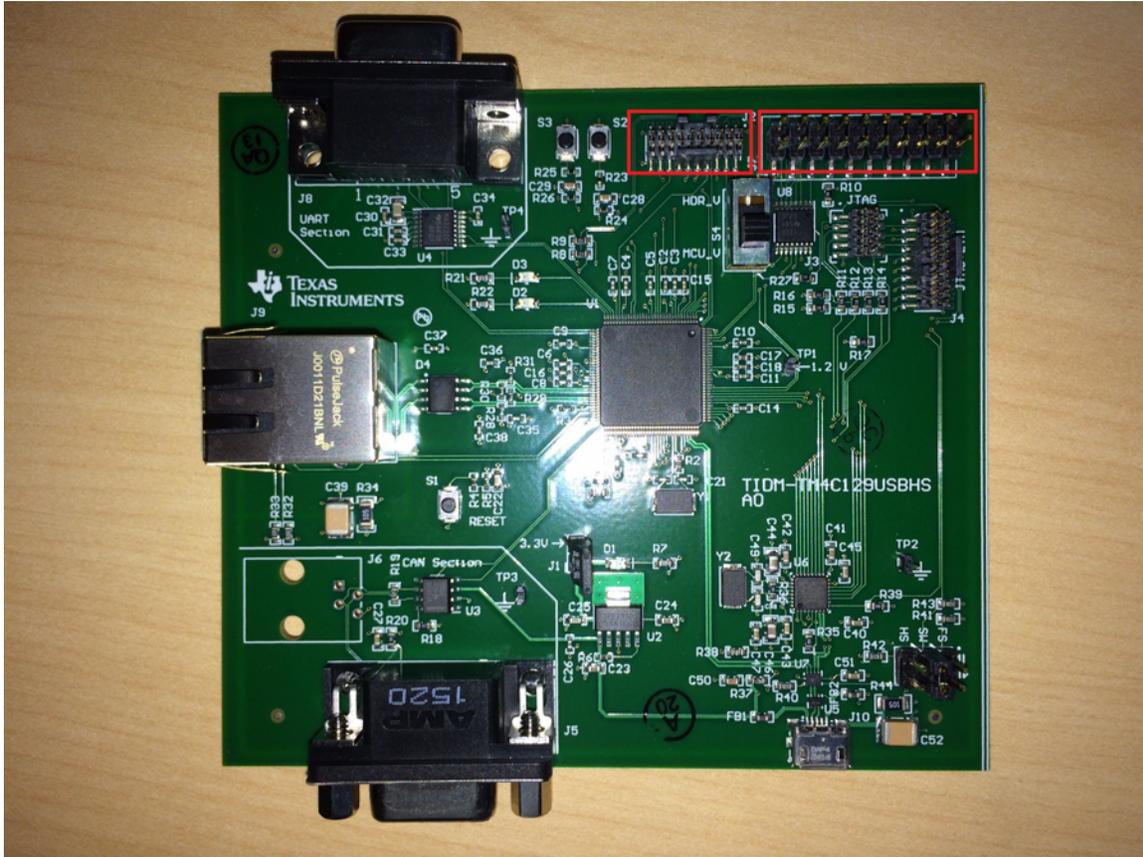


Figure 6. Low-Speed Interfaces

The J2 header allows the TM4C129x MCU to be connected to offboard devices or systems using UART, I²C, SSI, and ADC channels without any signal level shifting. [Table 2](#) gives the pin out of the low-speed interfaces available on the J2 header.

Table 2. J2 Header Pinout

J2 PIN	GPIO NAME	FUNCTION	J2 PIN	GPIO NAME	FUNCTION
1	3.3V	Power	2	PQ0	SSI3 CLK
3	PD5	UART2 TX	4	PQ1	SSI3 FSS
5	PD4	UART2 RX	6	PQ2	SSI3 XDAT0
7	PP1	SSI3 XDAT3	8	PA3	SSI3 XDAT1
9	PP0	SSI3 XDAT2	10	GND	Ground
11	PG1	I ² C1 SDA	12	GND	Ground
13	PG0	I ² C1 SCL	14	GND	Ground
15	PE2	ADC AIN1	16	PE3	ADC AIN0
17	PE0	ADC AIN3	18	PE1	ADC AIN1
19	GND	Ground	20	GND	Ground

The J6 header allows the TM4C129x MCU to be connected to offboard devices or systems using GPIO that may be bit banded for data or control through a level shifter. When the offboard devices or system are powered by a supply greater than 3.3 V, J6 Pin-1 does not need to be connected to the offboard power rail, and the switch S4 must be kept in the MCU_V position (see [Figure 7](#)). If the offboard devices are powered by a supply less than 3.3V, J6 Pin-1 must be connected to the offboard power rail, and the switch S4 must be kept in the HDR_V position (see [Figure 7](#)).

[Table 3](#) gives the pin out of the low speed interfaces available on the J6 header.

Table 3. J6 Header Pinout

J6 PIN	GPIO NAME	FUNCTION	J6 PIN	GPIO NAME	FUNCTION
1	PWR	Power	2	NC	No connect
3	NC	No connect	4	GND	Ground
5	PD0	MCU GPIO0	6	GND	Ground
7	PD1	MCU GPIO1	8	GND	Ground
9	PD2	MCU GPIO2	10	GND	Ground
11	NC	No connect	12	GND	Ground
13	PD3	MCU GPIO3	14	GND	Ground
15	PN0	MCU GPIO4	16	GND	Ground
17	PN1	MCU GPIO5	18	GND	Ground
19	NC	No connect	20	GND	Ground

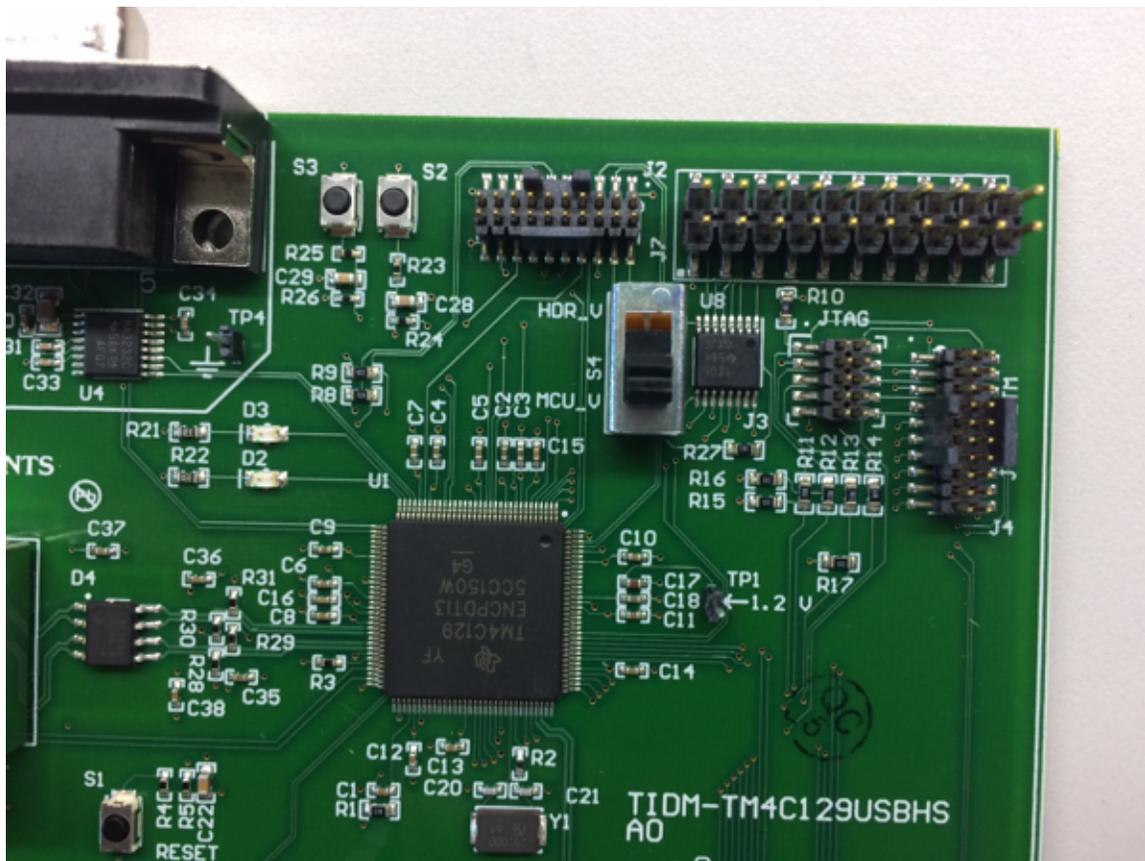


Figure 7. S4 Switch Position

2.1.4 Debug Interfaces

The reference design also features a standard 10-pin, 50-mil pitch ARM header (J3 header) and a 20-pin 50-mil pitch Cortex-M ETM header (J4 header) for debug, trace, and development of source code. When using the headers the user must ensure that only one of the headers has a debugger pod connected to it.

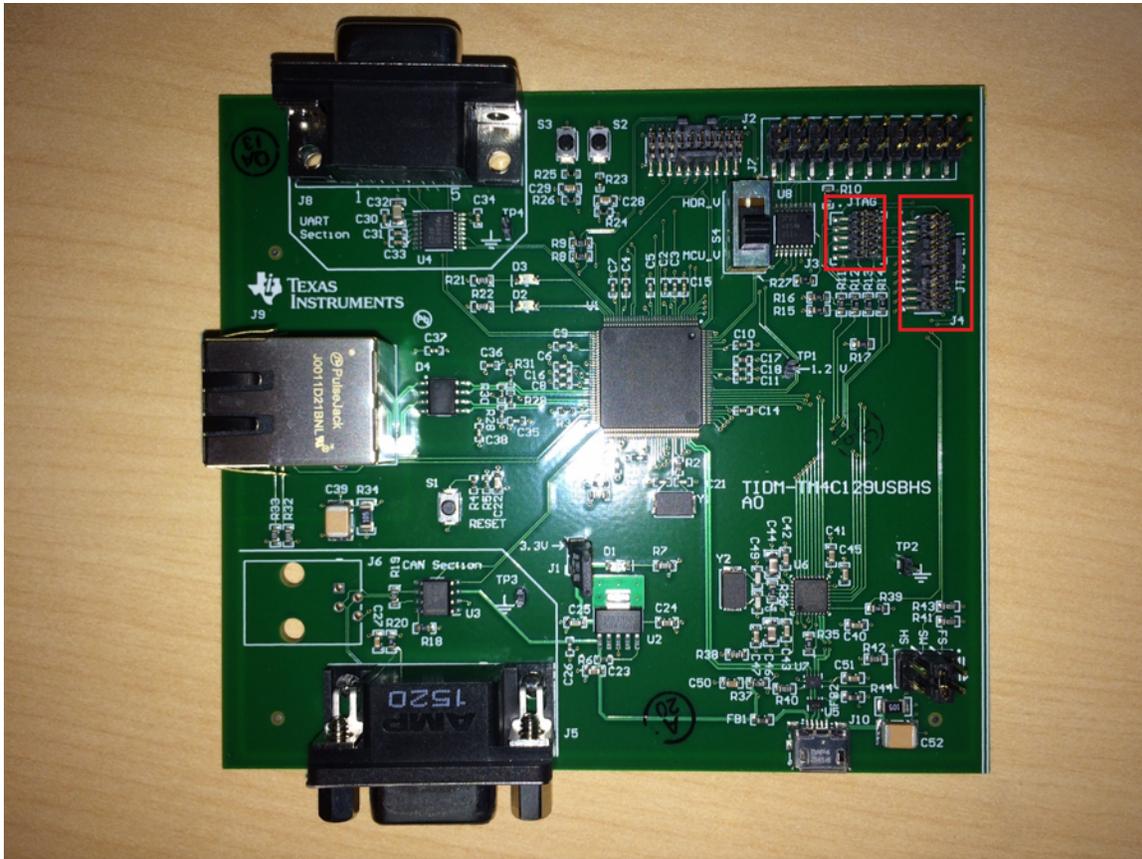


Figure 8. JTAG and ETM Header

2.1.5 Switches and LED

The board also features three push button switches, and two LEDs for user control and status as described in [Table 4](#).

Table 4. Switch and LED Connectivity

SWITCH AND LED	TYPE	GPIO
S1	Switch with onboard pull up	MCU reset pin RST_N
S2	Switch with onboard pull up	PC7
S3	Switch with onboard pull up	PC6
D2	LED	PH2
D3	LED	PH3

2.1.6 Power Options

The reference design is USB bus powered. When the user connects the USB cable, the 5 V from the USB is converted to the 3.3-V MCU power using an onboard LDO. Alternatively the user may directly supply 3.3 V by uninstalling the header J1 and connecting the pin marked 3.3 V to a bench power supply. A GND point is provided close to the J1 header when using the bench power supply. When running the example code for the reference design, the J1 header is installed and a header shunt connected so that the power is derived from USB connector.

2.2 Software

To run the example code and application, the following tools are required to be installed on the user PC.

- Download and install TivaWare 2.1.3.
- Download and install Code Composer Studio™ (CCS) 6.1.1.00022 with ARM compiler tool chain version 5.2.7
- Download and install LMFlashProgrammer build 1613.
- Download and install Uniflash 3.4.00003

2.2.1 Getting Started Software

The software for this reference design comes as an installer that the user must install on their PC. It normally takes about a minute for the installer to execute. When the installer is executed with the default settings, the software will be installed under *C:\Program Files (x86)\Texas Instruments\TM4C\TM4C129USBHS-1.0*. There are 3 directories that shall be created by the installer in the above path, which is explained below.

2.2.1.1 Directory appdrivers

The appdrivers directory contains the driver files built on top of TivaWare for high level application functions.

2.2.1.2 Directory tidm_tm4c129cusbhs2ethernet

The tidm_tm4c129cusbhs2ethernet directory contains the CCS project for interfacing the USBHS port to the Ethernet port.

2.2.1.3 Directory USBHSAApp

The USBHSAApp directory contains the Visual Studio project for the USB application that must be executed on the Windows® PC for this reference design. The user may modify the same to add their own code or change the layout of the Windows application.

2.2.1.4 Executable USBHSAApp.exe

The USBHSAApp.exe is the unmodified PC executable that must be run on the Windows PC for this reference design.

2.2.2 Example Application Description

The example application provided with the TI reference design performs the following tasks

1. Configures switch S2 press to force the device in ROM boot loader mode when the switch is pressed at the time of power up or when the reset switch S1 is pressed and released. This enables firmware upgrade without having to connect a debug probe to the design.
2. Allows switching between USBFS and USBHS by monitoring the GPIO state change when switch S3 is pressed at the time of power up or when the reset switch S1 is pressed and released. By default if the switch is not pressed, the bus is enumerated as a high-speed USB bus, and LED D2 is lit. If the switch is pressed, the bus is enumerated as a full speed-bus, and LED D3 is lit.
3. Once the USB bus is enumerated by the PC, the board can connect to the PC application, which can get the speed of the USB bus.

4. The Ethernet protocol stack connects to the network and acquires an IP address, from a DHCP server, which is displayed in the PC application.
5. The local PC can send data to a remote PC through the application example support. The user provides the remote board's IP address in the USBHSAApp PC application along with the message to send. On sending the message the local board relays the message to the remote board through the IP address provided by the local PC. The remote device then uses USB bulk transfer mode to send the data to the remote PC, which is running another instance of the USBHSAApp application.

2.2.3 Data exchange packet structure

A simple packet structure (shown in [Figure 9](#)) is used to exchange both data packets with the local and remote device and control packet with the local device. The first 4 bytes of the packet contains a pattern that is unlikely to occur in a packet to inform the local device if it is a request for data exchange with remote device or a request for control exchange with the local host.

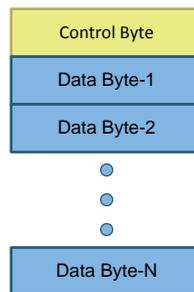


Figure 9. Packet structure

If the control byte contains the value 0x66554433, it is treated as a control packet request between the local host and the local device to exchange information. In the current example, there is no payload and the user may define the data bytes for appropriate information to be sent to the local device from the local host.

The local device shall always respond to the control packet with a response packet having the control byte as 0x99AABBCC. The data byte-1 is the payload and contains the IP address of the local device. When sending the data to the remote device, the control byte contains the IP address of the remote device and data bytes contain the actual data to be sent to the remote host.

2.2.4 Import and Build the Project in Code Composer Studio™

Before building the examples the user must create a new workspace to ensure a clean build of the project.

1. Start CCS.
2. Click on *File*, then in the drop down menu navigate to *Switch Workspace* → *Others*.
3. In the pop up box, provide the name of the new workspace. As an example we have created a workspace called *workspace_TIDMTM4C129USBHS* (see [Figure 10](#)). Click *OK*.

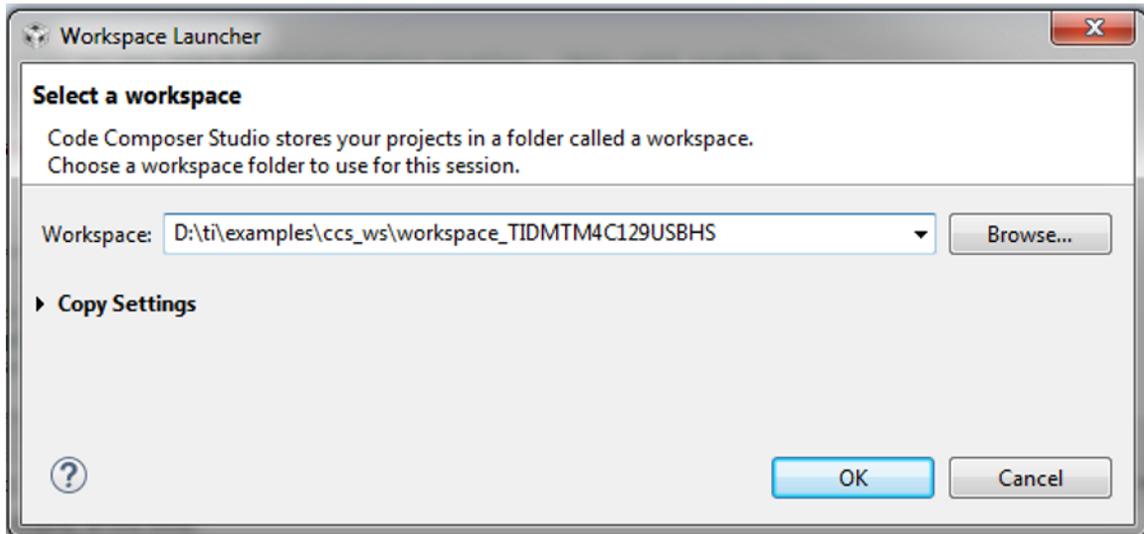


Figure 10. Creating a new workspace

4. In the workspace created, select *File* → *Import*. The *Import* window will be displayed.

- Expand CCS for more options then click CCS Projects. Click on *Next* (see [Figure 11](#)).

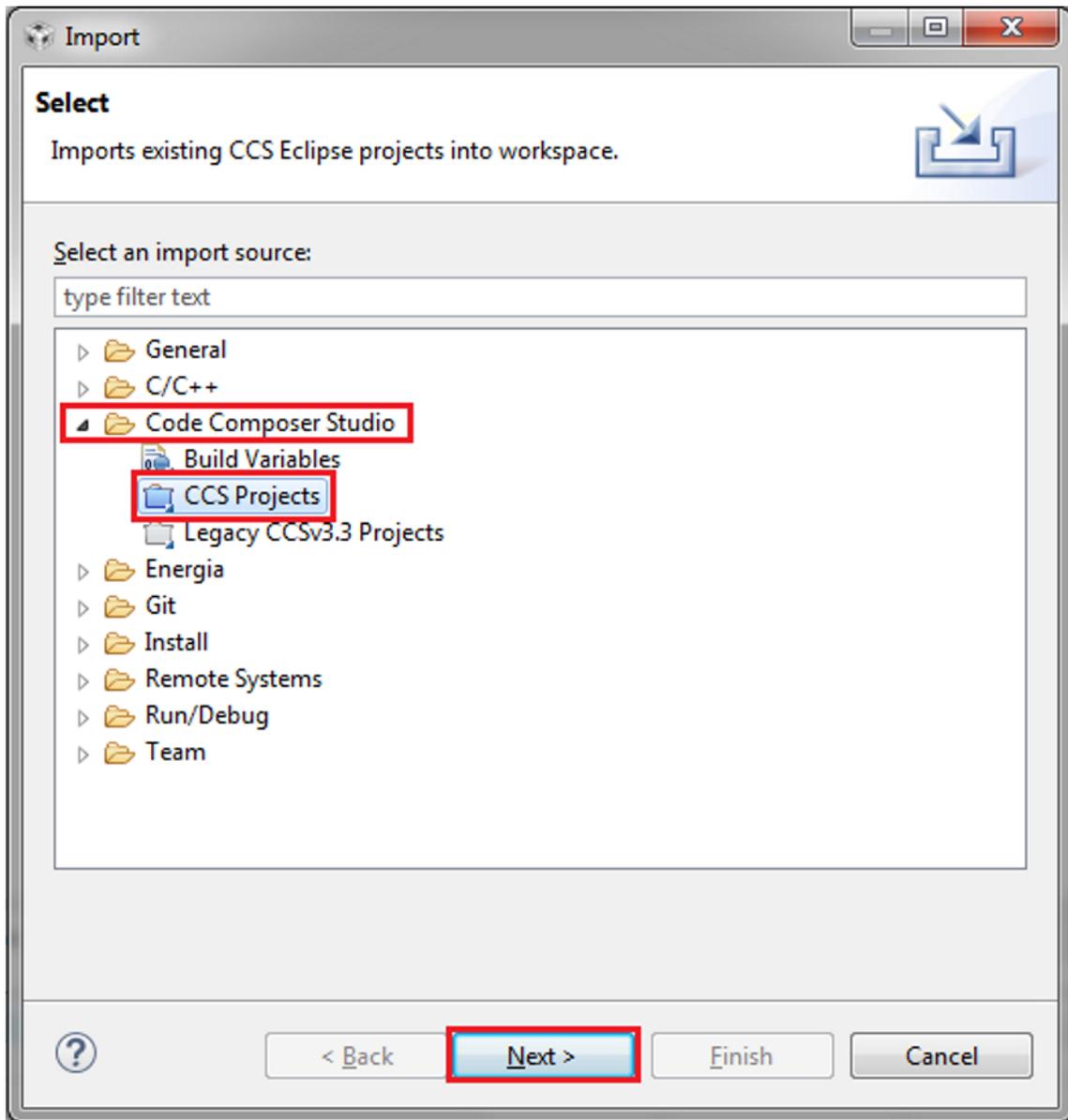


Figure 11. Import Selection View

- Click the *Browse* button in front of *Select search-directory* option, and navigate to the directory where the project collateral has been installed (see [Section 2.2.1](#)). Select all the projects that are listed in the *Discovered projects:* pane. Make sure that the check-boxes in front of the options *Automatically import referenced projects found in the same search-directory* and *Copy projects into workspace* are checked (see [Figure 12](#)). Now click on the *Finish* button. This will import the examples into CCS.

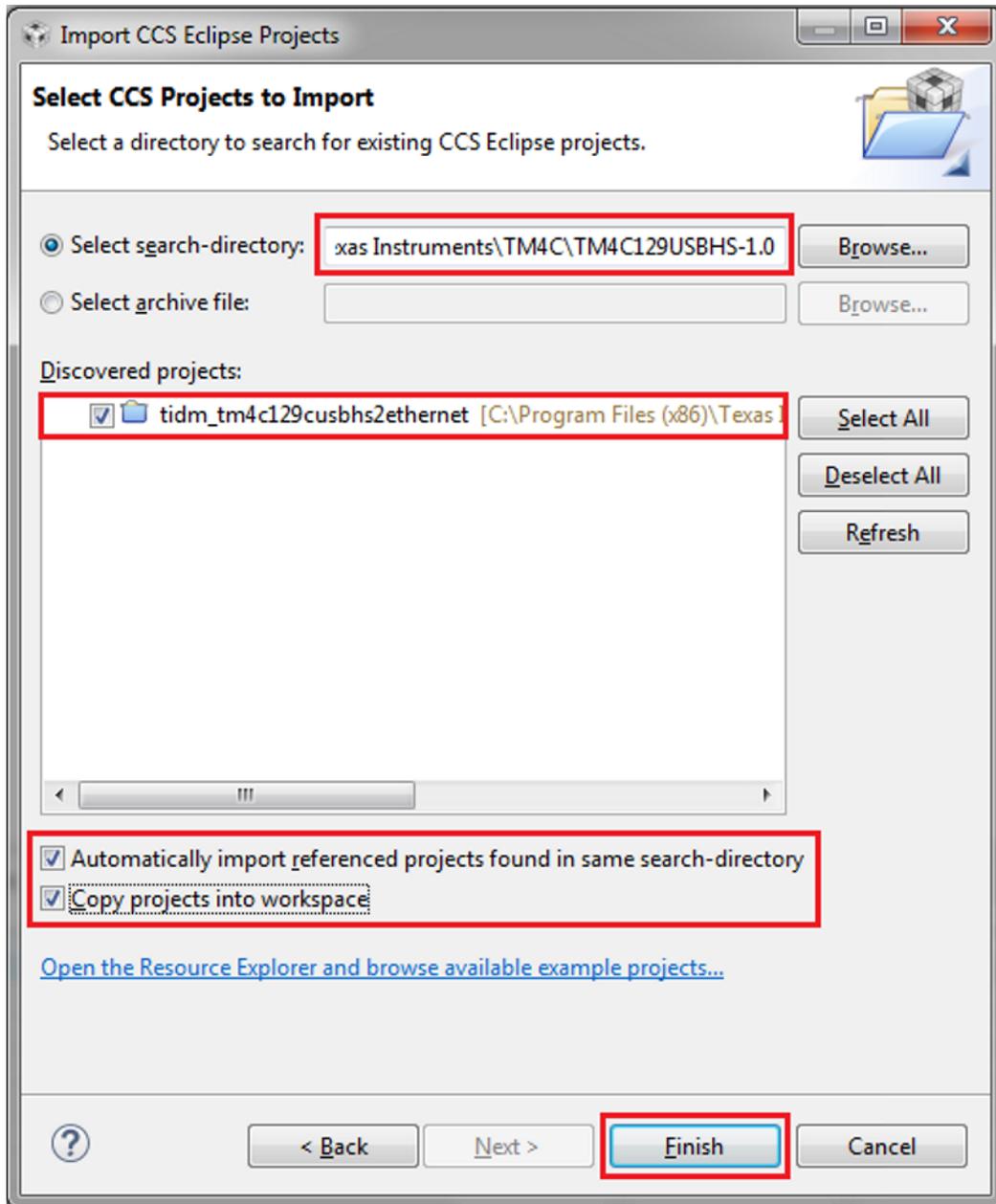


Figure 12. Import the Projects

- Before building the projects, ensure the variable SW_ROOT points to the correct directory where the TivaWare 2.1.3 has been installed and that TID_ROOT points to the correct directory where the project collaterals have been installed. To view and/or change the variable, right-click on the project and click on *Show Build Settings*. In the pop-up window, click on *Build*. Click on the *Variables* tab to view the value of SW_ROOT and/or TID_ROOT. If the path shown is different compared to the path on the machine, where the opus source code has been extracted, select the variable and click on *Edit...* to change the path (see Figure 13).

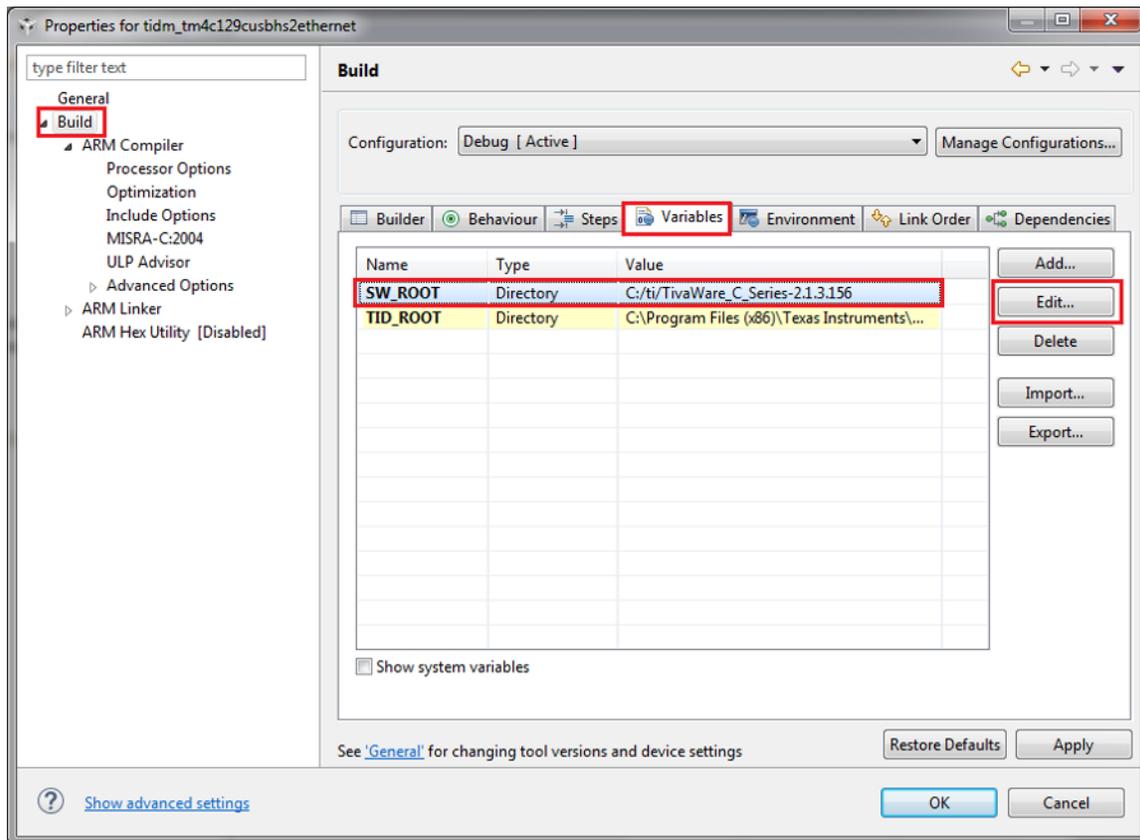


Figure 13. Update the Environment Variables

- Right-click the project in the Project Explorer and click *Build Project*. If the building the library for the first time, it may take a few minutes. After the compilation is successful, the CCS console must display the following message (see [Figure 14](#)).

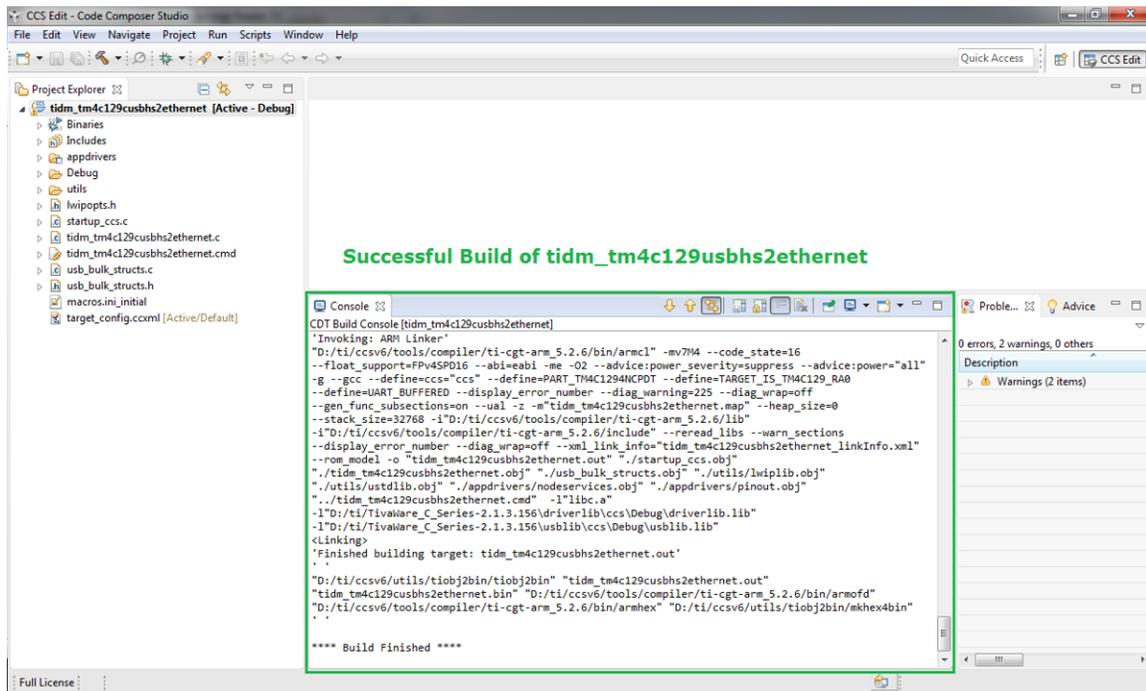


Figure 14. Compiling tidm_tm4c129usbhs2ethernet Project

2.2.5 Test Setup

The test setup consists of two TIDM-TM4C129USBHS boards. Board-1 is connected to PC-1 through a USB cable, and the Ethernet jack is connected to an Ethernet network switch. Board-2 is connected to PC-2 through another USB cable, and the Ethernet jack is connected to the same Ethernet network switch (as shown in [Figure 15](#)).

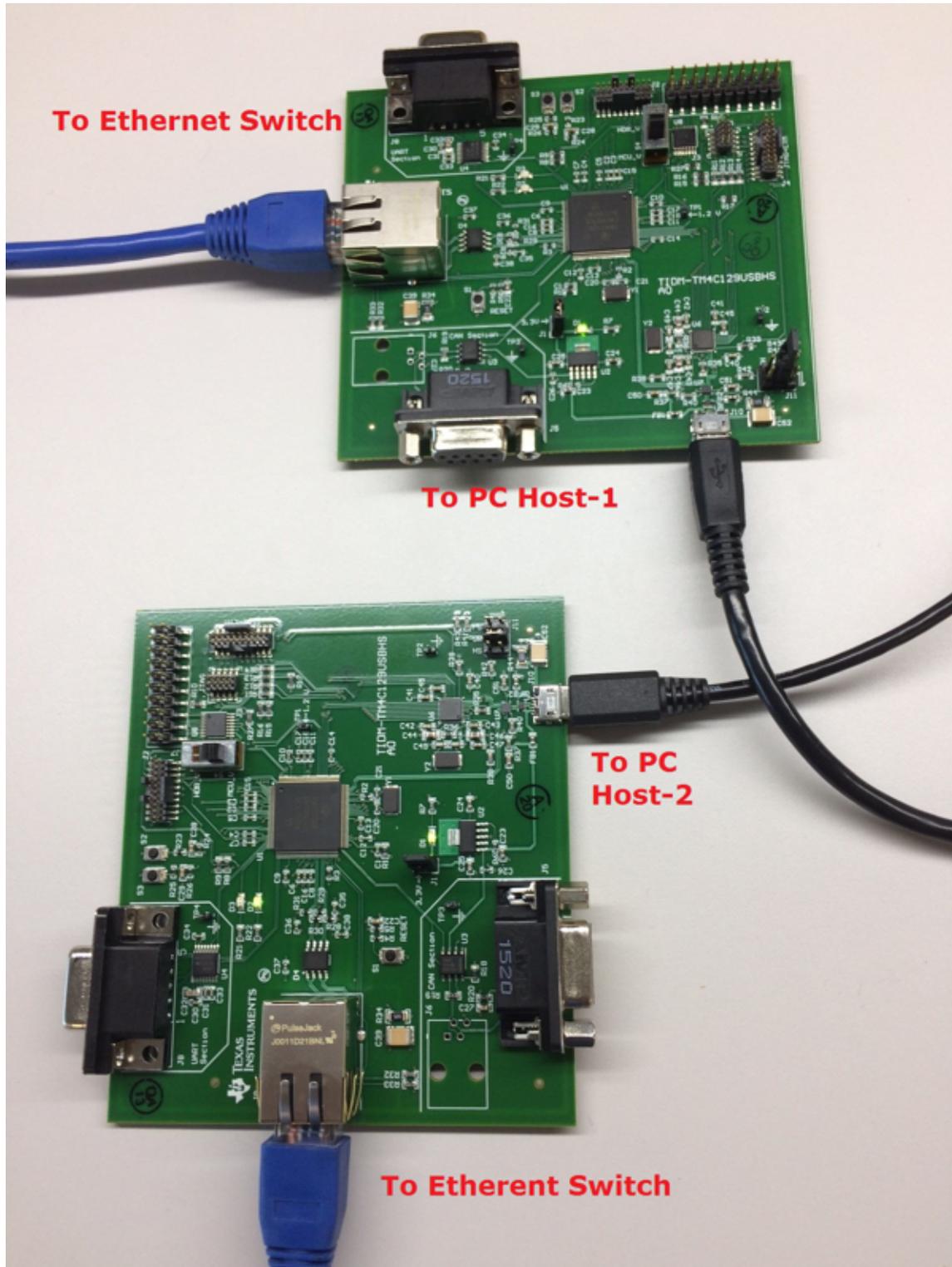


Figure 15. Test Setup

2.2.6 Executing the Example Code

On a factory reset part the flash and all configurations are in erased state. The first step is to connect a debug probe, like the XDS100, to the board and program the user registers 0 and 1 of the TM4C129x MCU. The value programmed in the user registers are used to program the MAC address of the Ethernet MAC controller.

1. Connect a XDS100 probe to the JTAG header J3 and power up the board by connecting the USB cable to a PC.
2. Start Uniflash application on the PC.
3. Click on *File* then *Open Configuration* in the drop down menu. In the pop-up navigate to the target configuration file *target_config.ccxml* (see Figure 16), which is provided as part of the project collateral installation. Press *OK*.

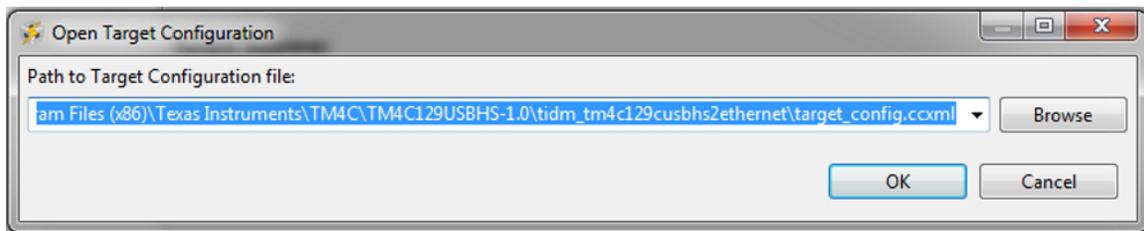


Figure 16. Loading Uniflash Configuration

4. Once the target configuration file is loaded, expand *Flash Settings* and click on *User Register Programming* option. In the main window, go to *MAC Address Mode*, and program the MAC address!- as per your block allocation of MAC address (see Figure 17). Ensure that the checkbox in front of *Commit MAC Address* is selected before pressing the *Program MAC Address* button.

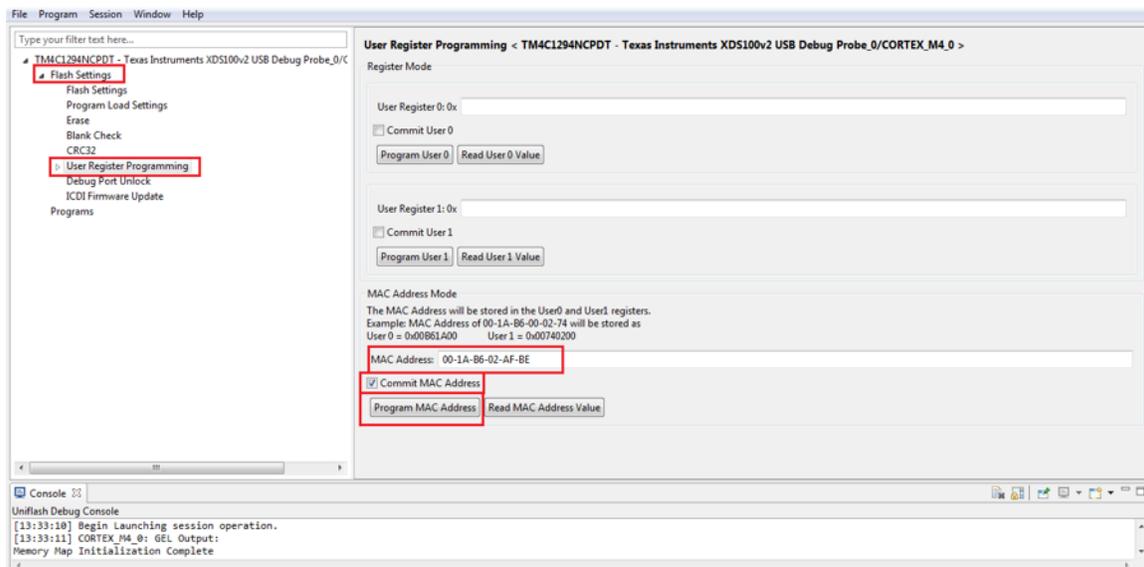


Figure 17. Programming the MAC Address

5. Disconnect the debug probe from the JTAG header as the TM4C129x default boot loader will be used to load the application code.

6. Launch the LMFlashProgrammer Application. In the *Configuration* tab, select *USB DFU* in the *Interface* drop-down menu (see [Figure 18](#))

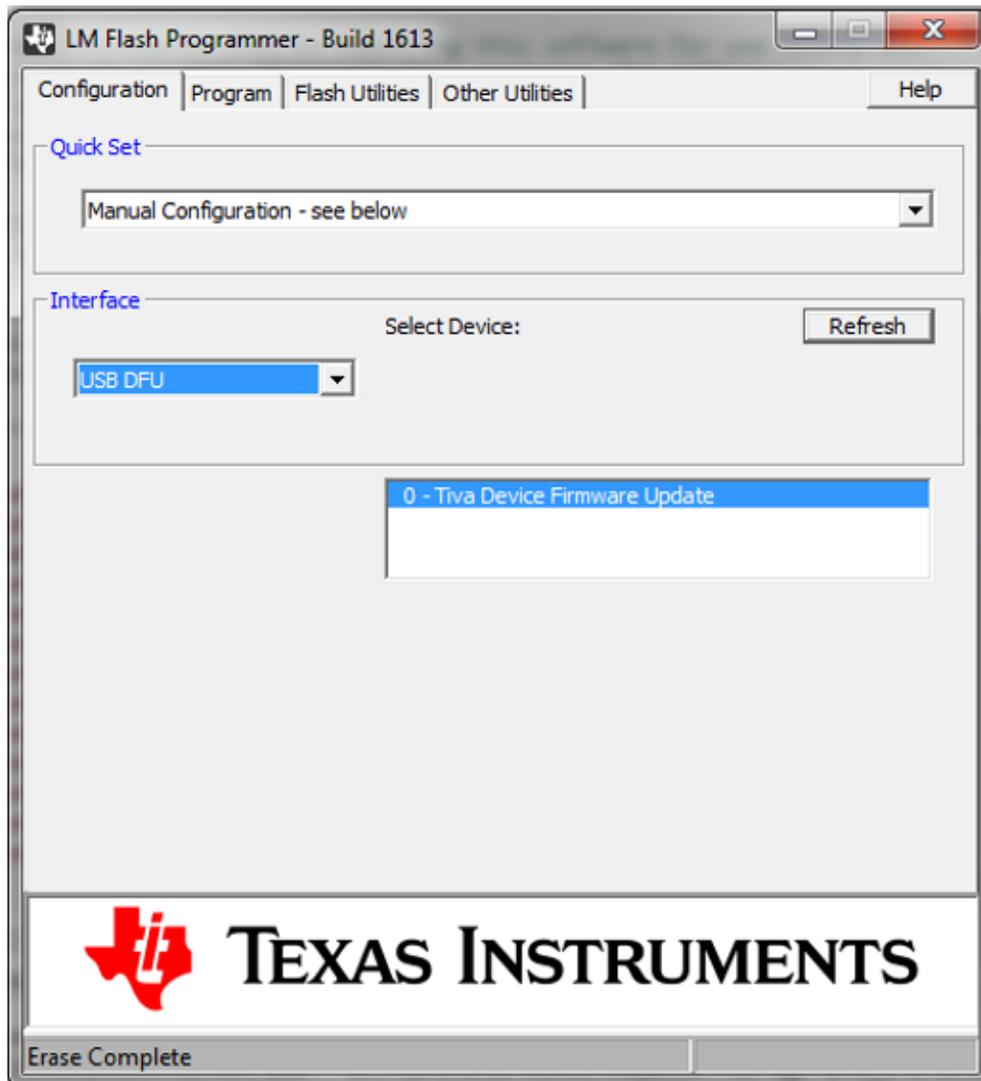


Figure 18. Selecting USB Interface for Application Download

- Click on the *Program* tab and select the binary file (see [Section 2.2.3](#)) generated in the CCS workspace (see [Figure 19](#)). Make sure the program address is set to 0x0 as shown in [Figure 19](#)

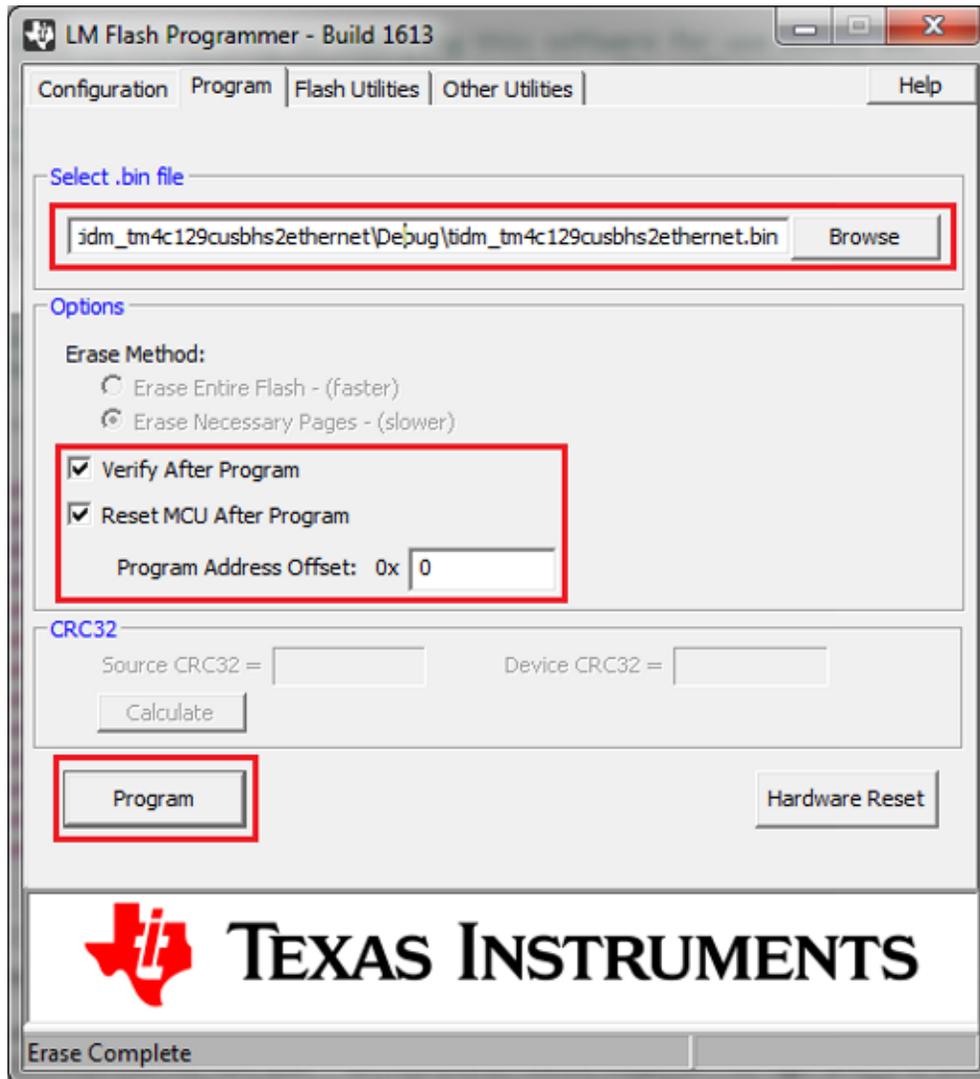


Figure 19. Programming the Application

8. Open Windows device manager. The board should get enumerated as a *TivaWare Bulk Device* in Windows device manager (see [Figure 20](#)).

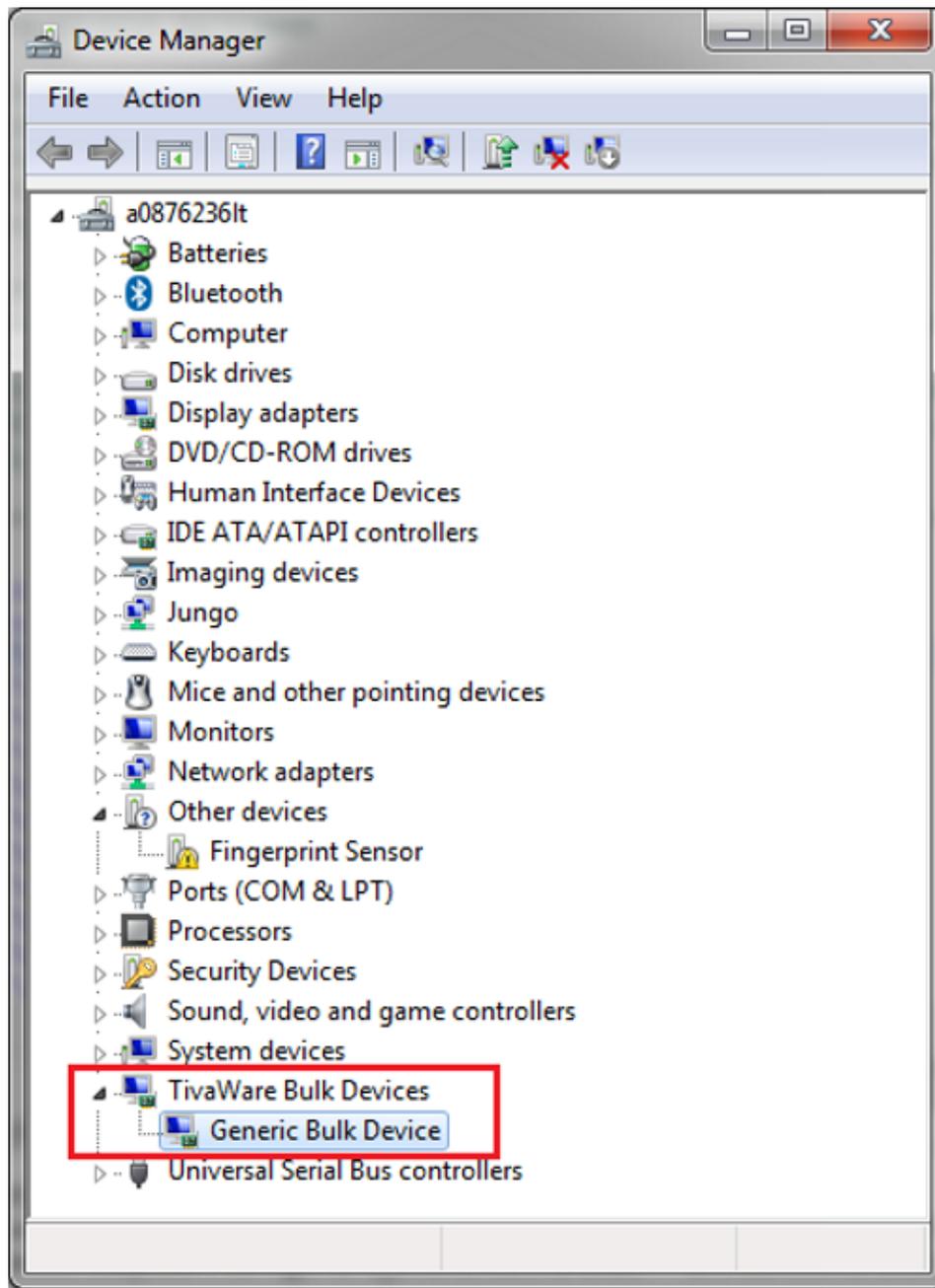


Figure 20. Windows View of the Device

- The device has now been successfully enumerated. Now connect to the device using the USBHSAApp provided in the project collaterals. Make sure that the Ethernet cable is connected to the RJ45 jack on the board and the switch. Start the USBHSAApp on the local Windows machine. Click on the *USB Connect* button (see [Figure 21](#)).

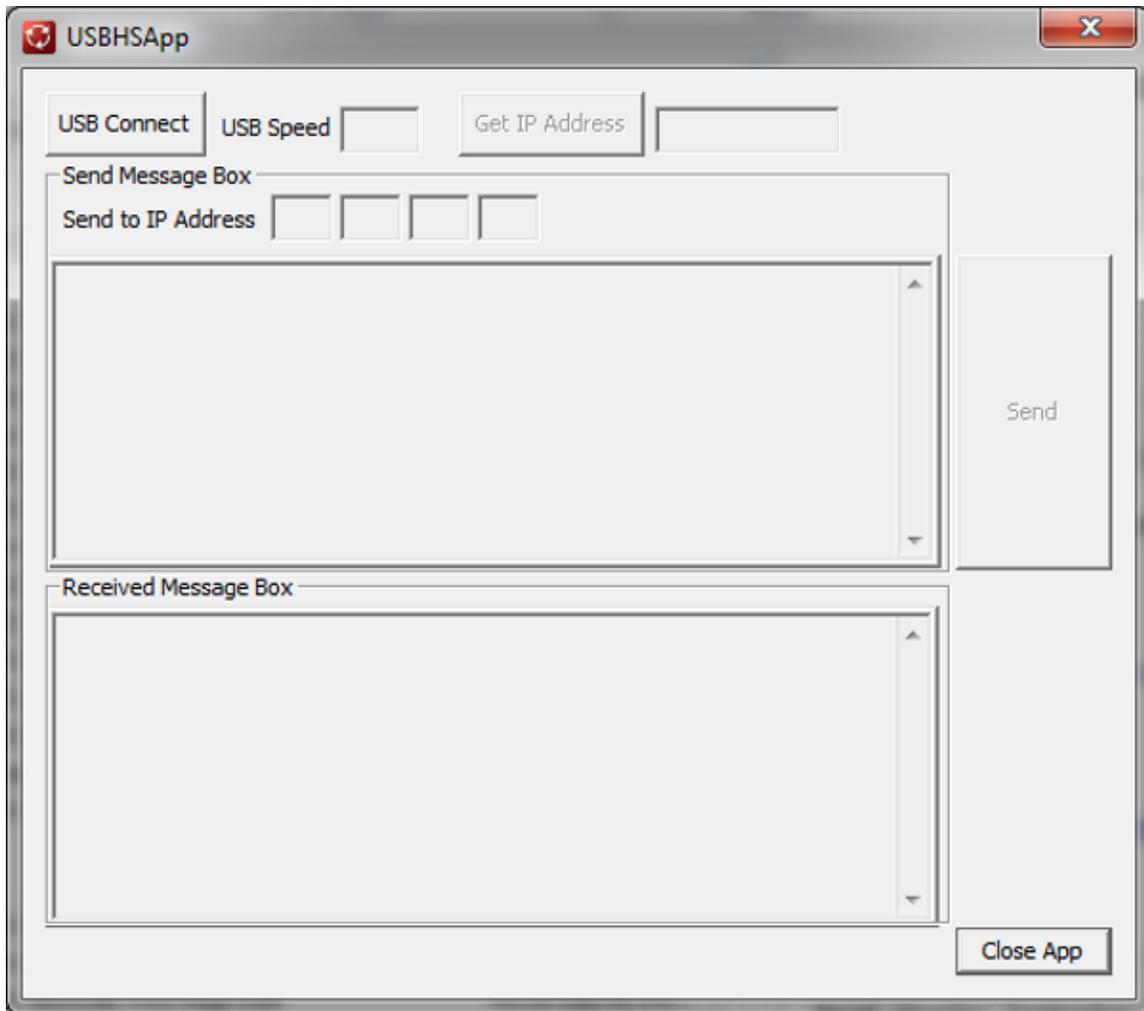


Figure 21. Running the USBHSAApp

10. As the device has been enumerated already, the USB Speed edit box shall now show *HS* as the operating mode and also enable the other buttons in the application (See [Figure 22](#)). If the switch S3 is pressed when the board is powered up or the board is reset then the USB Speed edit box shall show *FS* as the operating mode.

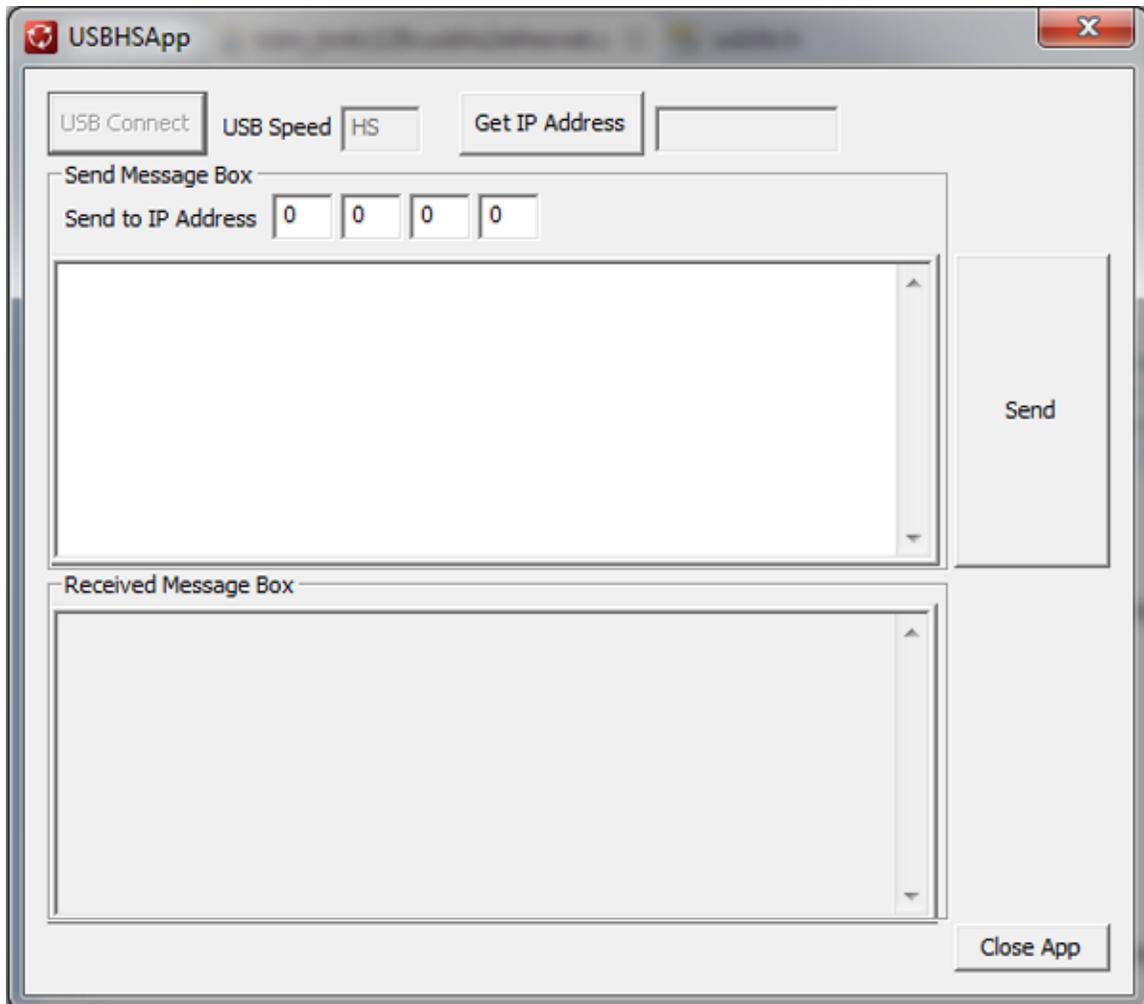


Figure 22. Running the USBHSAApp

- Click on the button *Get IP Address*, which will cause the application to send a request to the local device to send the IP address of the local device on the network (see [Figure 23](#)). If the Ethernet cable is not plugged in, then it will show *Link not established*. If the Ethernet cable is plugged, but the IP address has not been assigned, then it will show *Acquiring IP Address*.

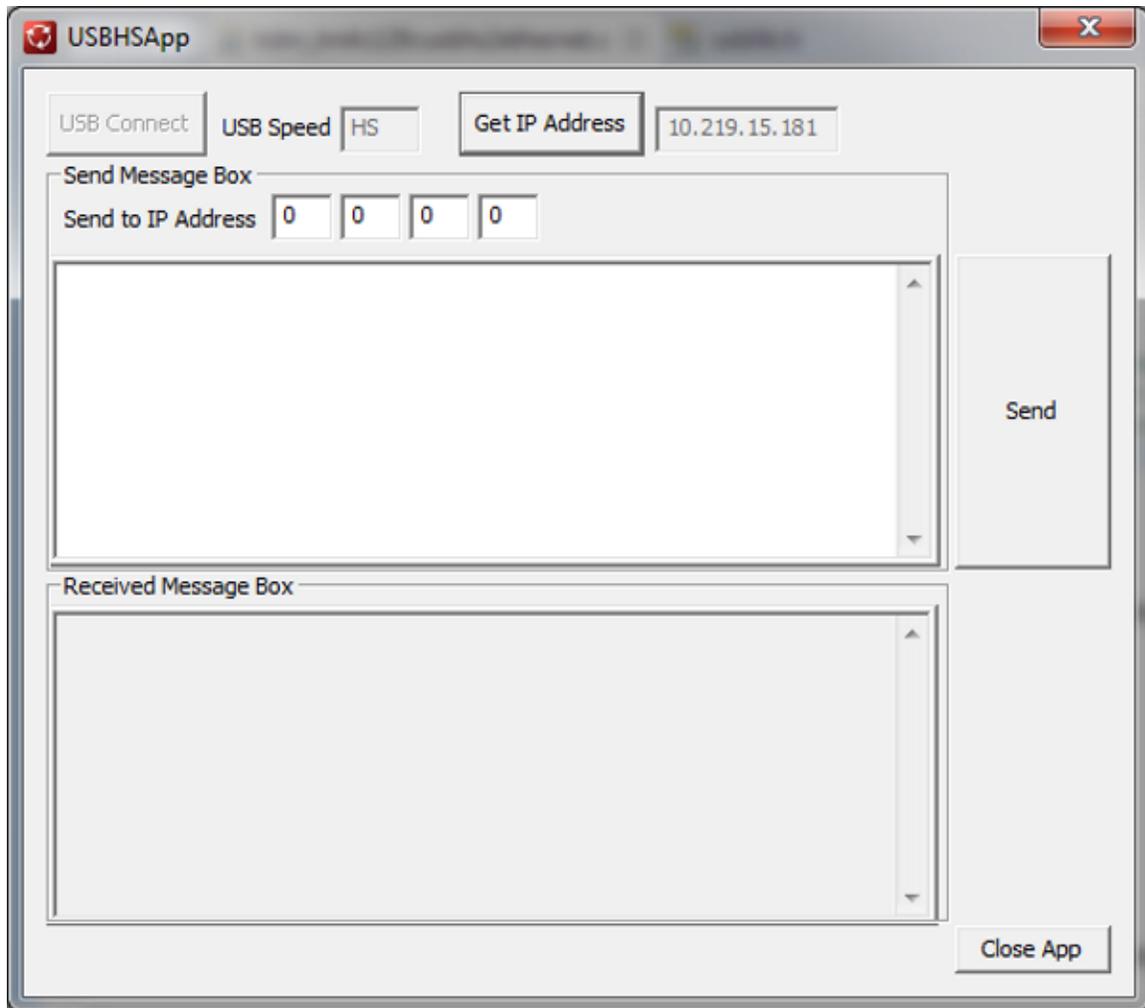


Figure 23. Display the IP Address of the Local Device

- Repeat the same process (steps 1 to 11) for the second board connected to another PC host (referred to as remote host).

- To send messages from the local host to the remote host, enter the IP address of the remote host in the *Send to IP Address* edit box of the local host. Now type the message in the *Send Message Box* and click the *Send* button. The message will be shown in the *Received Message Box* of the USBHSAApp running on the remote device (see [Figure 24](#)).

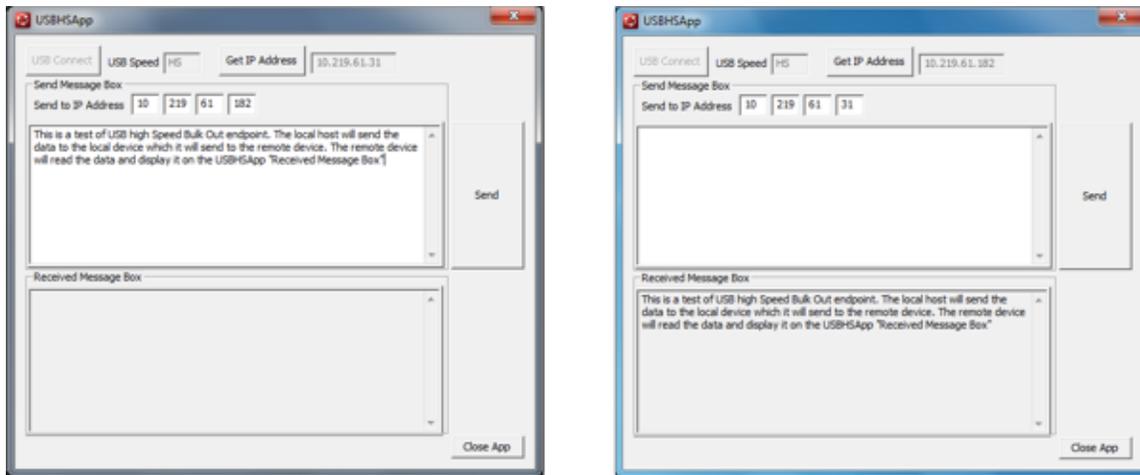


Figure 24. Send Message From Local to Remote Host

- To send messages from the remote host to the local host, enter the IP address of the local host in the USBHSAApp of the remote host. Now type the message in the *Send Message Box*, and click the *Send* button. The message will be shown in the *Received Message Box* of the USBHSAApp running on the local device (see [Figure 25](#)).

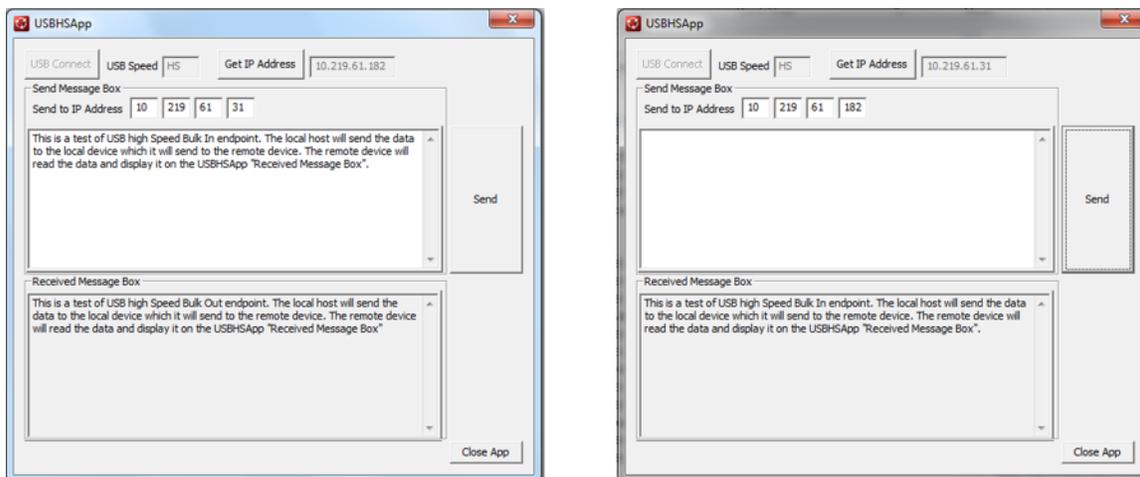


Figure 25. Send Message From Remote to Local Host

3 Testing and Results

A USB protocol analyzer has been connected during the testing of the design. When the application is run in USBHS mode, the USB protocol analyzer shows that the bus is configured as USB HS bus and the Start of Frame (SOF) comes every 125us. Also when data is sent from one board to the other, the maximum packet size of 512 bytes for bulk transfers is used as shown in the figures below.

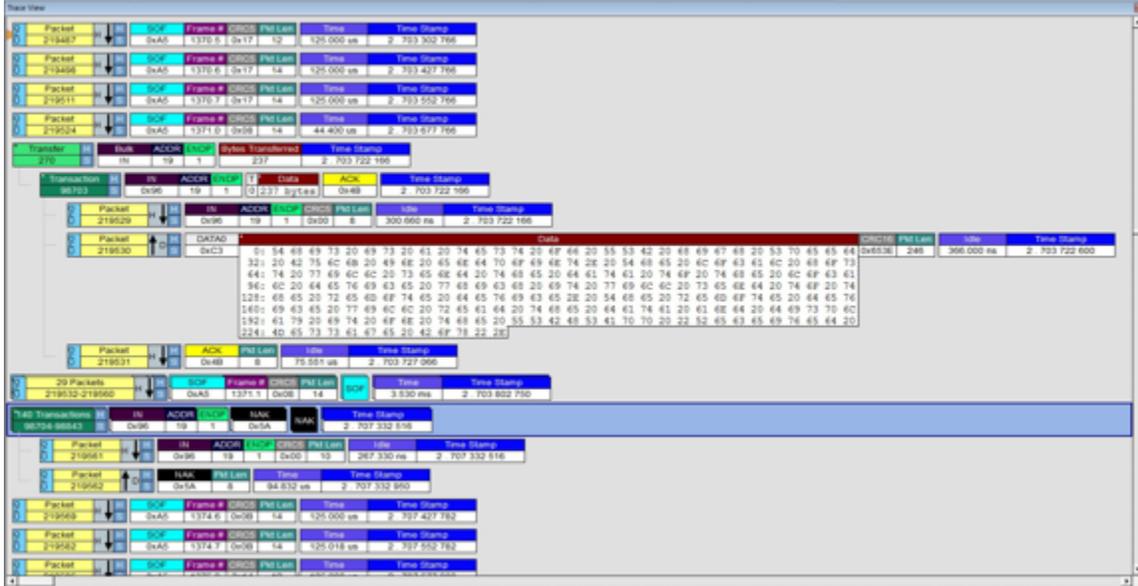


Figure 26. USBHS bulk IN transfer capture

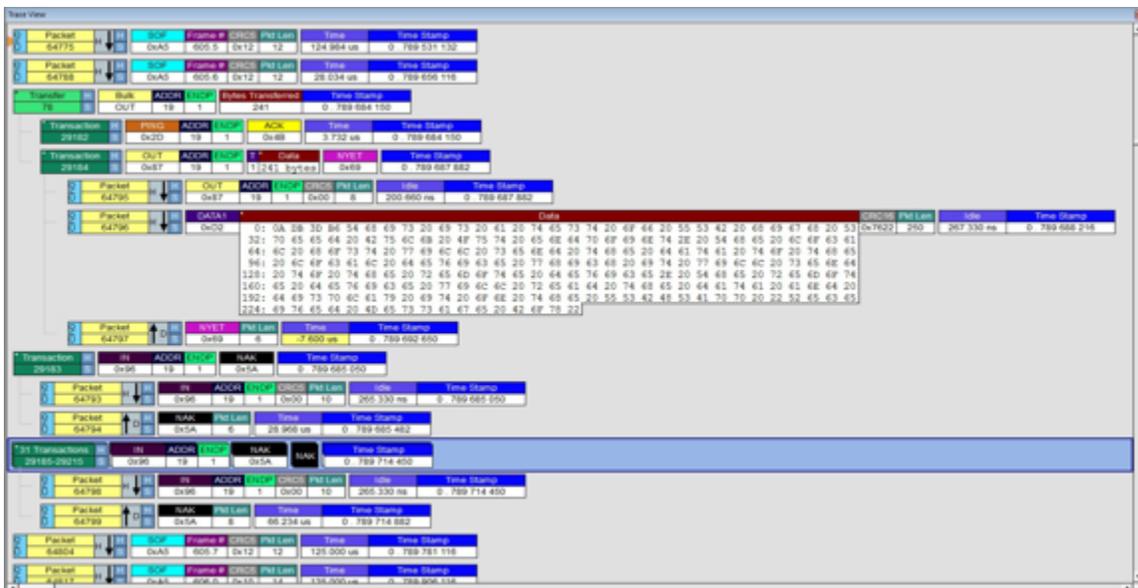


Figure 27. USBHS bulk OUT transfer capture

4 Design Files

4.1 Schematics

To download the schematics, see the design files at [TIDM-TM4C129USBHS](#).

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at [TIDM-TM4C129USBHS](#).

4.3 PCB Layout Recommendations

4.3.1 USB and Ethernet Analog Trace

An important consideration when doing the layout is the trace width for Ethernet and USB signals. The Ethernet and USB interfaces have critical differential impedance requirements. Both Ethernet signal pairs should be routed as a $100\ \Omega \pm 10\%$ differential pair on the top layer of the PCB with a ground plane as a reference. The USB signal pair should be routed as a $90\ \Omega \pm 10\%$ differential pair on the top layer of the PCB with a ground plane as a reference.

The most optimal solution is if the PCB fab house can adjust the stack up and provide for controlled dielectric. The designer should use the PCB tools to set the spacing and width of the traces to get close to the target characteristic impedance. The PCB fab house can then adjust the trace space and width to their specific materials and process.

During PCB layout, if the PCB fab house has a predefined layer stack up for low cost process, the user must ascertain the layer stack up information. Then use this information in PCB tools to get the optimum trace width. The design files have used a low-cost variant with the following PCB stack up for four layer PCBs.

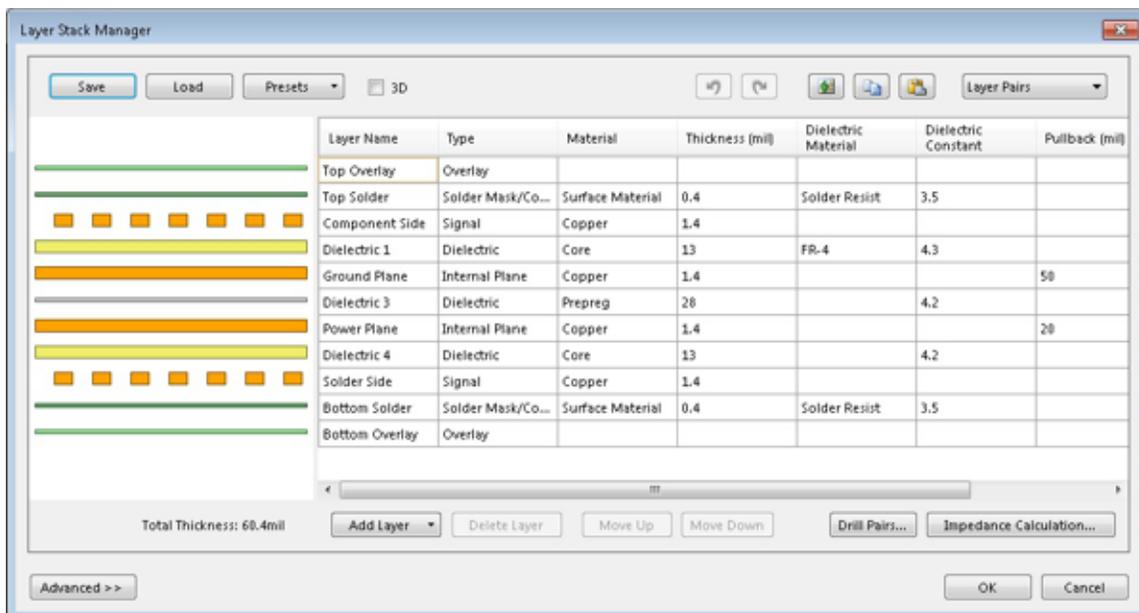


Figure 28. PCB Layer Stack up for TIDM-TM4C129USBHS

Table 5. Differential Signals Trace Information

TRACE WIDTH (mil)	TRACE THICKNESS (mil)	TRACE HEIGHT (mil)	TRACE SPACING (mil)	E_R	Z_{DIFF}	Z_0
10	0.4	15.8	5	4.2	109.476	84.766
12.8	0.4	15.8	5	4.2	99.336	76.915

4.3.2 USB-ULPI trace

The USB-ULPI bus is a 60-MHz, bidirectional data bus. The PCB layout has to be done optimally to ensure that the interface timings are met both at the MCU and the USBHS PHY. To achieve the same, the overall trace length of the USB-ULPI bus must not exceed 6 inches, and all USB-ULPI traces must be length matched within 500 mils of each other.

4.4 Altium Project

To download the Altium project files, see the design files at [TIDM-TM4C129USBHS](#).

4.5 Gerber Files

To download the Gerber files, see the design files at [TIDM-TM4C129USBHS](#).

4.6 Assembly Drawings

To download the assembly drawings, see the design files at [TIDM-TM4C129USBHS](#).

5 Software Files

To download the software files, see the design files at [TIDM-TM4C129USBHS](#).

6 Related Documentation

1. Texas Instruments, [Implementation of Programmer for Serial Bootloaders on TM4C12x Microcontroller](#), Application Report (SPMA074)
2. Texas Instruments, [Using TM4C12x devices over JTAG Interface](#), Application Report (SPMA075)

6.1 Trademarks

All trademarks are the property of their respective owners.

7 About the Author

AMIT ASHARA is an Application Engineer and Member Group Technical Staff at Texas Instruments where he works on developing applications for the TM4C12x family of high-performance microcontrollers. Amit brings his extensive experience in high-speed digital and microcontroller system-level design expertise to this role. Amit earned his Bachelor of Engineering (BE) from University of Pune, India.

IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

Texas Instruments Incorporated ("TI") reference designs are solely intended to assist designers ("Designer(s)") who are developing systems that incorporate TI products. TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.

TI's provision of reference designs and any other technical, applications or design advice, quality characterization, reliability data or other information or services does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such reference designs or other items.

TI reserves the right to make corrections, enhancements, improvements and other changes to its reference designs and other items.

Designer understands and agrees that Designer remains responsible for using its independent analysis, evaluation and judgment in designing Designer's systems and products, and has full and exclusive responsibility to assure the safety of its products and compliance of its products (and of all TI products used in or for such Designer's products) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to its applications, it has all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. Designer agrees that prior to using or distributing any systems that include TI products, Designer will thoroughly test such systems and the functionality of such TI products as used in such systems. Designer may not use any TI products in life-critical medical equipment unless authorized officers of the parties have executed a special contract specifically governing such use. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death (e.g., life support, pacemakers, defibrillators, heart pumps, neurostimulators, and implantables). Such equipment includes, without limitation, all medical devices identified by the U.S. Food and Drug Administration as Class III devices and equivalent classifications outside the U.S.

Designers are authorized to use, copy and modify any individual TI reference design only in connection with the development of end products that include the TI product(s) identified in that reference design. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY 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 products 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 the reference design or other items described above 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 AND OTHER ITEMS DESCRIBED ABOVE ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY DESIGNERS AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS AS DESCRIBED IN A TI REFERENCE DESIGN OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TI's standard terms of sale for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>) apply to the sale of packaged integrated circuit products. Additional terms may apply to the use or sale of other types of TI products and services.

Designer will fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer's non-compliance with the terms and provisions of this Notice.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2016, Texas Instruments Incorporated