Application Report
SPMA075–August 2016

# Using TM4C12x Devices Over JTAG Interface

Amit Ashara

## ABSTRACT

The IEEE Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture (JTAG) is a method for verifying designs and testing printed circuit boards after assembly. It is used as the primary means for transferring data to a nonvolatile memory of an embedded system and debugging embedded software.

This application report describes the physical connections for JTAG and design considerations to be taken into account for a custom board. It also shows how to use the JTAG interface on the TM4C12x LaunchPad™ for debugging the on-board microcontroller using an external debugger, or by using the on-board debugger for debugging an off-board microcontroller.

**NOTE:** This document applies to TM4C12x Series microcontrollers. All screen captures reflect the TM4C12x devices.

### Contents

### List of Figures

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

**List of Tables**

# 1 Introduction

JTAG defines a Test Access Port (TAP) and Boundary Scan Architecture for digital integrated circuits and provides a standardized serial interface for controlling the associated test logic. It can be used to test the interconnections of assembled printed circuit boards and obtain manufacturing information on the components. It also provides a means of accessing and controlling the design for test features such as I/O pin observation and control, scan testing, and debugging.

The JTAG port on TM4C12x is comprised of four pins:

- Test Clock (TCK)
- Test Mode Select (TMS)
- Test Data In (TDI)
- Test Data Out (TDO)

Even though the TM4C12x devices support Serial Wire Debug (SWD) mode, this application report focuses on JTAG, which is a more widely adopted interface.

# 2 Overview of JTAG Protocol

Before the interfacing of TM4C12x devices over JTAG can be discussed, it is important to understand the basic concepts of JTAG protocol and terminologies. This aids the debugging of system issues when JTAG does not work as expected.

## *2.1 JTAG State Machine*

JTAG accesses the Test Access Port (TAP) of a device by changing TMS and TDI in conjunction with TCK and reading results through TDO.

- TDI and TMS are sampled on the rising edge of TCK by the TAP.
- TDO is changed on the falling edge of TCK by the TAP.

The JTAG works by accessing the Instruction Register (IR) and the Data Register (DR). The IR is a 4-bit serial scan chain connected between the TDI and TDO pins. When the TAP controller is in the correct state, bits can be shifted into the IR. Once the IR is loaded, they are decoded to get access to the DR. The DR format is specific to the register being accessed. As an example:

- When the IR is loaded with the BYPASS instruction, the DR length is a 1-bit shift register.
- When the IR is loaded with IDCODE instruction, the DR length is a 32-bit shift register, which on shift out, gives the JTAG ID of the device that is specific to the manufacturer name, part number and version of the ARM core.

At power on, the TAP state machine (see Figure 1) is initialized to be in Test Logic Reset state. It moves from one state to another based on the TMS value (shown as logic 0 or 1 on the transition arrow) with every TCK. Once the state machine enters the Shift state, the TDI pin is used to serially shift in the IR or DR. At the same time, the value captured in the shift register during the Capture state is shifted out on to the TDO. When the Update state is executed, the value shifted in during the Shift state is updated to the TAP for the next JTAG cycle.

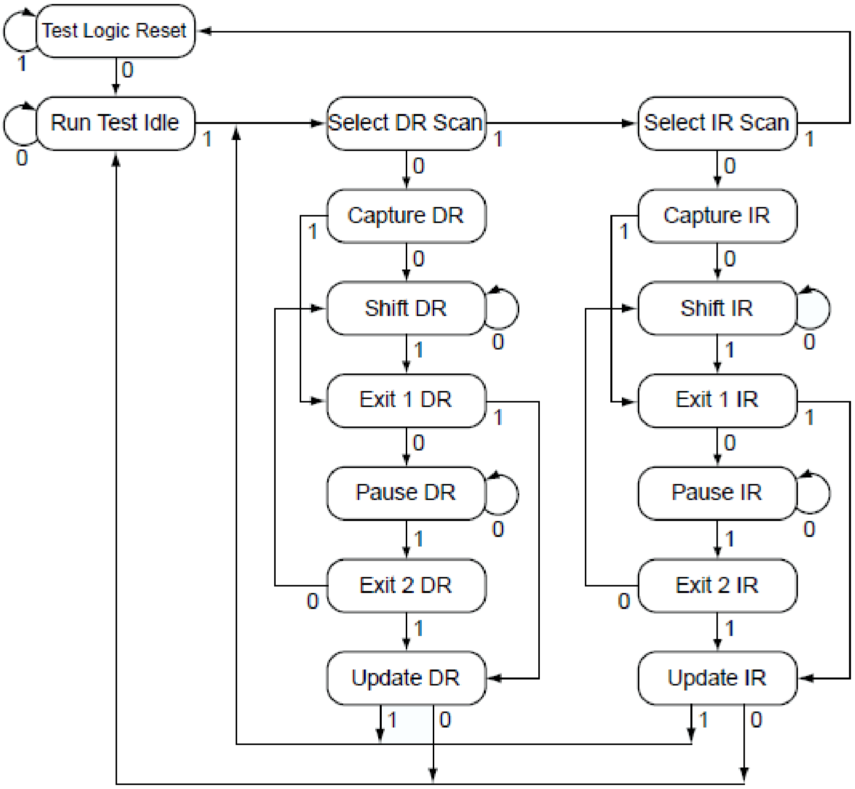


**Figure 1. Test Access Port State Machine**

## 2.2 JTAG System Implementations

A system with JTAG can be implemented in two different methods: point-to-point connection and daisy chain connection.

### 2.2.1 Point-to-Point Connection

Using a point-to-point method, a single set of JTAG pins are connected to a single device (see Figure 2). If there are multiple devices in a system, then multiple JTAG headers are required. The advantage of the point-to-point connection is that the speed of the device access is higher when compared to the daisy chain connection (discussed in Section 2.2.2). However, this increases the BOM cost and is more difficult to manage when performing cross trigger debug.

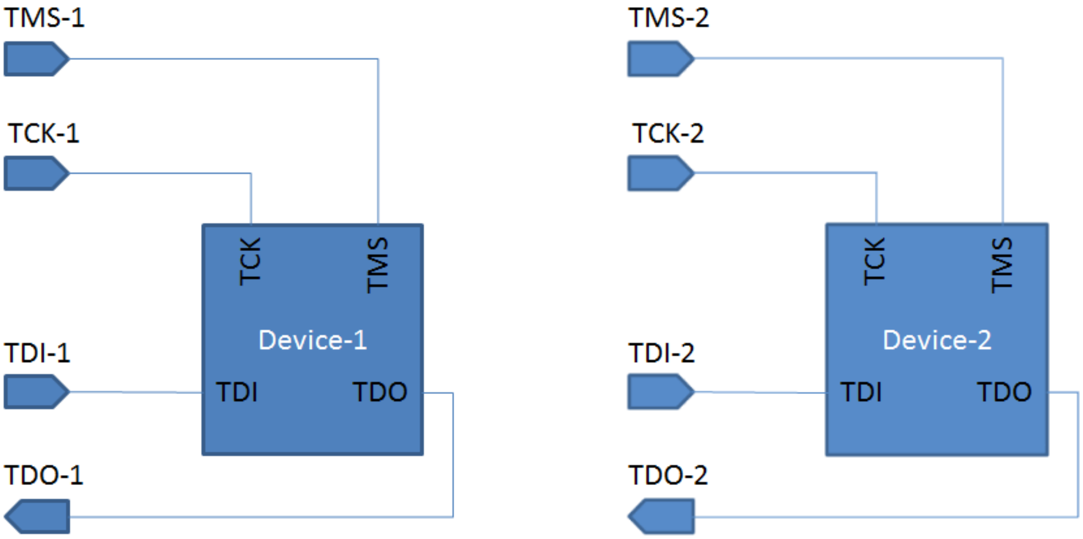


**Figure 2. Point-to-Point Connection**

### 2.2.2 Daisy Chain Connection

Using the daisy chain connection method, a single set of JTAG pins are connected to multiple devices (see Figure 3). The TDI from the JTAG header is connected to the TDI of the first device. The TDO of the first device is then connected to the TDI of the next device and so on until the last device. The TDO of the last device is connected to the TDO on the JTAG header. TCK and TMS are shared between all the devices. The advantage of a daisy chain connection is that the BOM cost is lower as individual headers per device is not required, PCB is simpler to layout and cross trigger debugging is convenient. Since multiple device TAPs are in the path, JTAG access is slower as additional shifts are required to bring each device TAP to the correct state.

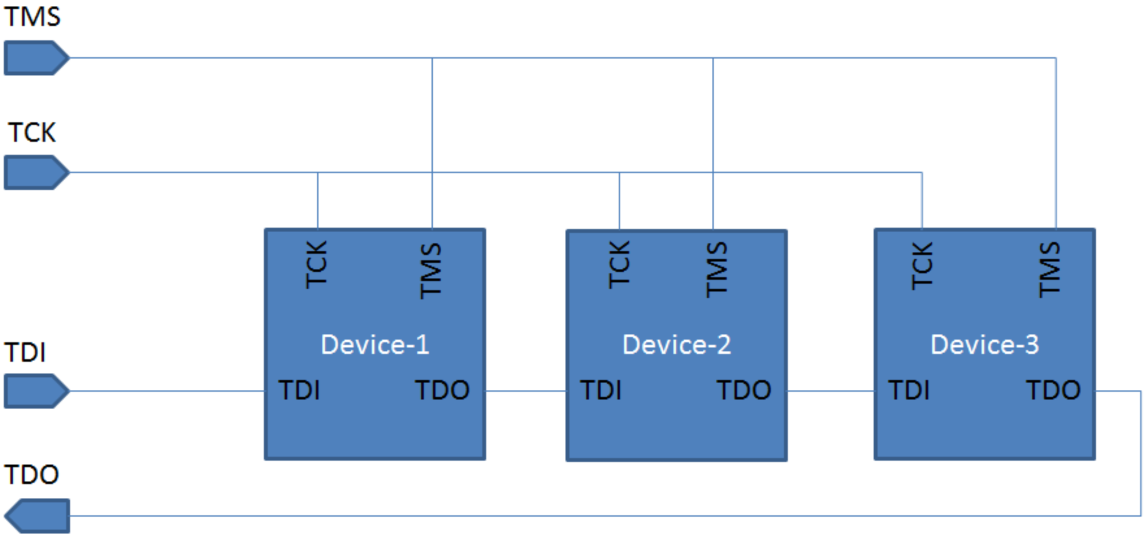


**Figure 3. Daisy Chain Connection**

## 2.3 *Additional JTAG Information*

Besides the four JTAG pins, there are two more pins available on some devices: TRST and RTCK.

- TRST: Test Reset. This pin is an optional pin and can be used to reset the JTAG TAP state machine. It is an active low signal.
- RTCK: Return TCK. This is a clock that is sourced by the device. When available, the TDO is sampled with this clock allowing for higher JTAG operation frequency. This is also referred to as adaptive clocking.

**NOTE:** TRST and RTCK are unavailable on all TM4C12x devices.

# 3 JTAG Debug Probes

To access the JTAG interface on an embedded system requires JTAG debug probes. These are often referred to as emulators (though this is a misnomer). The JTAG debug probes have control logic that generates the JTAG signaling and comes with drivers that a PC-based application like IDE's and programmers can utilize to download a firmware or debug an application.

## 3.1 JTAG Header Pin Out

The JTAG debug probe connects to headers on the PCB. There are different types of JTAG connectors and even though they serve the same function, it can be confusing when making the selection of the debug probe. To be able to make the correct connection, it is necessary to first understand the pin mapping of the JTAG header pin out. Table 1 shows the pin out for the different type of headers.

**Table 1. JTAG Header Pin Out**

| Pin Number | TI 14-Pin | Compact TI 20-Pin cTI | ARM 10-Pin | ARM 20-Pin |
|---|---|---|---|---|
| | 100 mil pitch | 50 mil pitch | 50 mil pitch | 100 mil pitch |
| 1 | TMS | TMS | VTRef | VTRef |
| 2 | nTRST | nTRST | TMS | VSupply |
| 3 | TDI | TDI | GND | nTRST |
| 4 | TDIS | TDIS | TCK | GND |
| 5 | VTRef | VTRef | GND | TDI |
| 6 | KEY | KEY | TDO | GND |
| 7 | TDO | TDO | KEY | TMS |
| 8 | GND | GND | TDI | GND |
| 9 | RTCK | RTCK | GNDDetect | TCK |
| 10 | GND | GND | nRESET | GND |
| 11 | TCK | TCK | | RTCK |
| 12 | GND | GND | | GND |
| 13 | EMU0 | EMU0 | | TDO |
| 14 | EMU1 | EMU1 | | GND |
| 15 | | nRESET | | nRESET |
| 16 | | GND | | GND |
| 17 | | EMU2 | | NC |
| 18 | | EMU3 | | GND |
| 19 | | EMU4 | | NC |
| 20 | | GND | | GND |

## 3.2 Connecting Debug Probes to TM4C12x

The previous section lists the different pinout based on the connector selected. However, based on the function of each pin, it may still be unclear what needs to be connected to the TM4C12x for debug and what to do with unused pins. Table 2 describes the function of each of the pins listed earlier in Table 1 and how they need to be connected to TM4C12x devices.

**Table 2. Pin Connectivity**

| Pin Name | TM4C12x Pin to connect | Comments |
|---|---|---|
| TCK | PC0 pin with a 10K pull up | Test Clock |
| TMS | PC1 pin with a 10K pull up | Test Mode Select |
| TDI | PC2 pin | Test Data In |
| TDO | PC3 pin with a 10K pull down | Test Data Out |
| VTRef | VDD supply of TM4C12x via a 100 Ω series resistor | Reference voltage used by level shifter on the debug probe to level shift signals if the IO voltage on the embedded system and debug probe are different |
| VSupply | NC | Not Used |
| TDIS | GND | Target disconnect detect |
| RTCK | NC | Return Test Clock for adaptive clocking |
| EMU0 | NC | Function depends on the target device |
| EMU1 | NC | Function depends on the target device |
| EMU2 | NC | Reserved for future use |
| EMU3 | NC | Reserved for future use |
| EMU4 | NC | Reserved for future use |
| KEY | NC | Pin removed from the header, so that the debug probe can be connected on un-shrouded header |
| nTRST | NC | Test Reset pin |
| nRESET | RST_N pin with a 10K pull up resistor | Target Reset pin |
| GNDDetect | GND | Same as TDIS |
| GND | GND | Common Ground |

**NOTE:** The pull up values mentioned in Table 2 are only for guidance. If there is a strong noise source close to the JTAG traces or reset pin, then the pull up resistor value needs to be decreased to make it less susceptible to noise coupled from traces.

## 3.3 JTAG Debug Probes and Software

There are multiple vendors that supply standalone JTAG debug probes like Spectrum Digital and Black Hawk to name two. Besides the standalone debug probes, TI embedded LaunchPads and EVMs (DK-TM4C123G and DK-TM4C129x) come with an on board debug probe. Also, TI provides IDE and standalone programming software like LMFlashProgrammer and Uniflash to download firmware. This section briefly introduces some of the debug probes that can be interfaced with TM4C12x and software support.

### 3.3.1 XDS100 Series

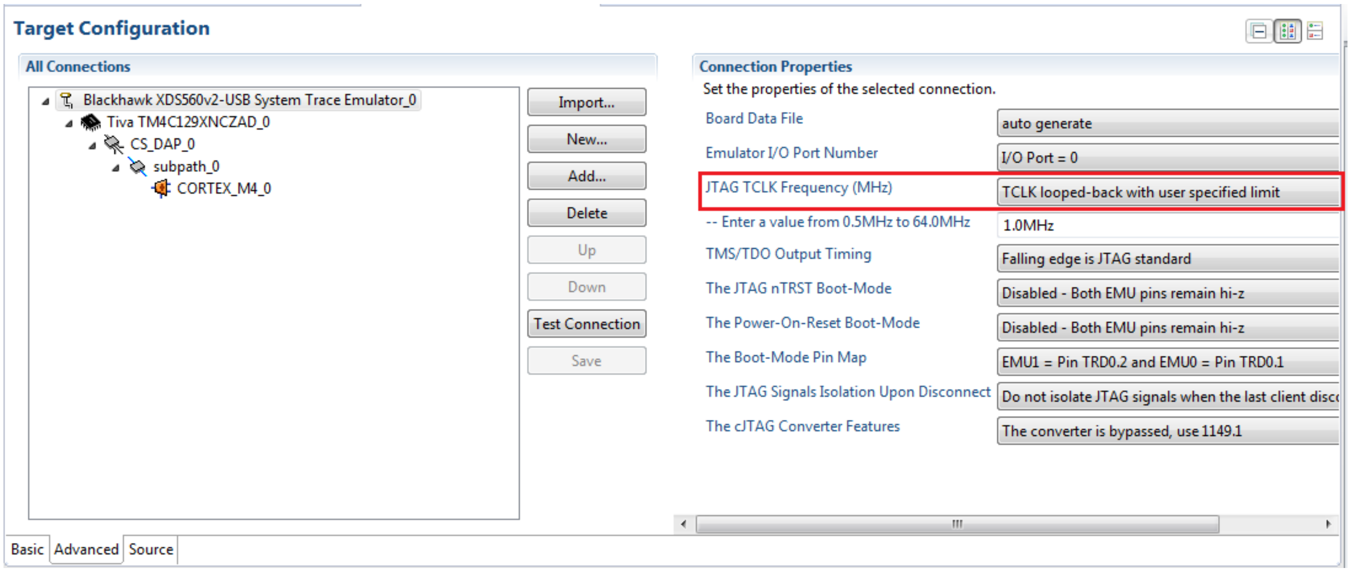
XDS100 is the entry level debug probe and is provided by multiple vendors. TM4C is supported by both v2 and v3 versions of XDS100 series.

### 3.3.2 XDS200

XDS200 is an intermediate debug probe that has better performance than XDS100 and lower cost than XDS560v2.

### 3.3.3 XDS560v2

XDS560v2 is a high performance debug probe and is provided by multiple vendors. When setting XDS560v2 as the debug probe, ensure that the Connection Properties has the option “JTAG TCLK Frequency (MHz) set to “TCLK looped-back with a user-specified limit” (see Figure 4).



Target Configuration

All Connections

- Blackhawk XDS560v2-USB System Trace Emulator_0
  - Tiva TM4C129XNCZAD_0
    - CS_DAP_0
      - subpath_0
        - CORTEX_M4_0

Connection Properties

Set the properties of the selected connection.

| | |
|---|---|
| Board Data File | auto generate |
| Emulator I/O Port Number | I/O Port = 0 |
| JTAG TCLK Frequency (MHz) | TCLK looped-back with user specified limit |
| -- Enter a value from 0.5MHz to 64.0MHz | 1.0MHz |
| TMS/TDO Output Timing | Falling edge is JTAG standard |
| The JTAG nTRST Boot-Mode | Disabled - Both EMU pins remain hi-z |
| The Power-On-Reset Boot-Mode | Disabled - Both EMU pins remain hi-z |
| The Boot-Mode Pin Map | EMU1 = Pin TRD0.2 and EMU0 = Pin TRD0.1 |
| The JTAG Signals Isolation Upon Disconnect | Do not isolate JTAG signals when the last client disc |
| The cJTAG Converter Features | The converter is bypassed, use 1149.1 |

**Figure 4. XDS560v2 Debug Probe Special Consideration**

### 3.3.4 Stellaris ICDI

The Stellaris ICDI is an on-board debug probe for the TM4C LaunchPads and EVMs. The on-board debug probe allows for a lower cost of the LaunchPads and EVMs therefore, a separate debug probe does not need to be procured. On the TM4C LaunchPad there is a provision for using an external debug probe or to use the Stellaris ICDI as a standalone debug probe.

### 3.3.5 I-jet

The I-jet debug probe is provided by IAR systems for IAR’s embedded workbench IDE.

### 3.3.6 ULINK2

The ULINK2 debug probe is provided by ARM for Keil uVision IDE.

### 3.3.7 Software Support for Debug Probe

It is also important to know which software to use when using a JTAG debug probe and what features are supported for TM4C12x. Table 3 is a useful look up when selecting a debug probe and programming software.

**Table 3. Software Support for Debug Probe**

| | IDE Supported | UniFlash Support | LMFlashProgrammer Support | Unlock Feature |
|---|---|---|---|---|
| XDS100 | Code Composer Studio | Yes | No | Yes, via UniFlash |
| XDS200 | Code Composer Studio | Yes | No | Yes, via UniFlash |
| XDS560 | Code Composer Studio | Yes | No | No |
| ICDI | Code Composer Studio<br>IAR Embedded Workbench<br>Keil uVision | Yes | Yes | Yes, via UniFlash and LMFlashProgrammer |
| IAR I-jet | IAR Embedded Workbench | No | No | No |
| uLink2 | Keil uVision | No | No | No |

## 4 Using JTAG Debug Probes With TM4C12x

After having a basic understanding of the JTAG protocol, debug probes and pin out, the next step is to interface the debug probe to a TM4C12x microcontroller. The following sections reference schematics for the different header pin outs.

### *4.1 Sample Schematic for TI 14-Pin Header*

Figure 5 shows how the JTAG header on the board needs to be connected to the TM4C12x device for a TI 14-pin header.

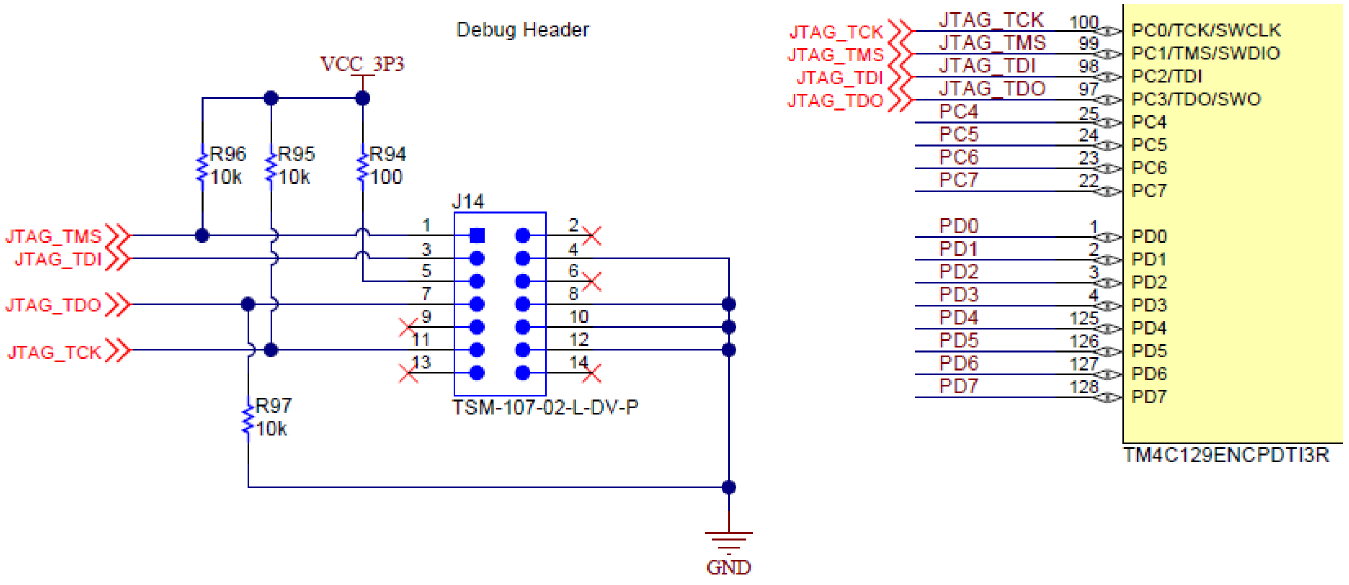


**Figure 5. TI 14-Pin Header**

### 4.2 Sample Schematic for Compact TI 20-Pin Header

Figure 6 shows how the JTAG header on the board needs to be connected to the TM4C12x device for a compact TI 20-pin header.

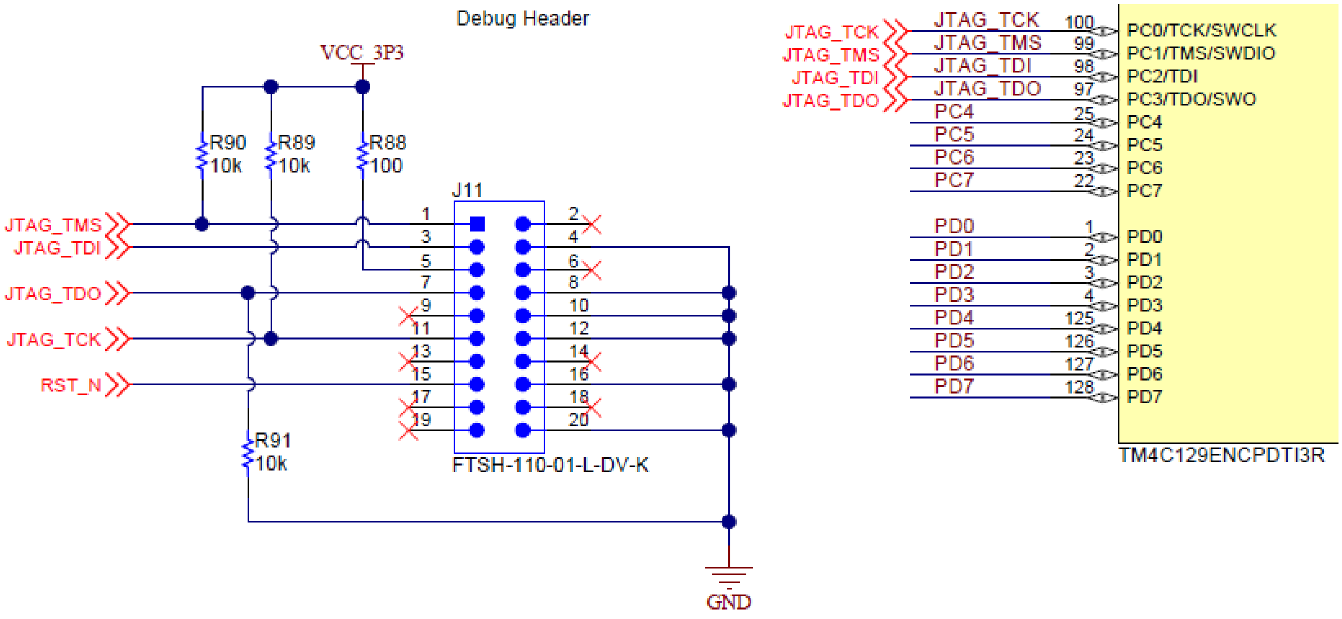


**Figure 6. CTI 20-Pin Header**

### 4.3 Sample Schematic for ARM 10-Pin (Cortex Debug) Header

Figure 7 shows how the JTAG header on the board needs to be connected to the TM4C12x device for an ARM 10-pin header.

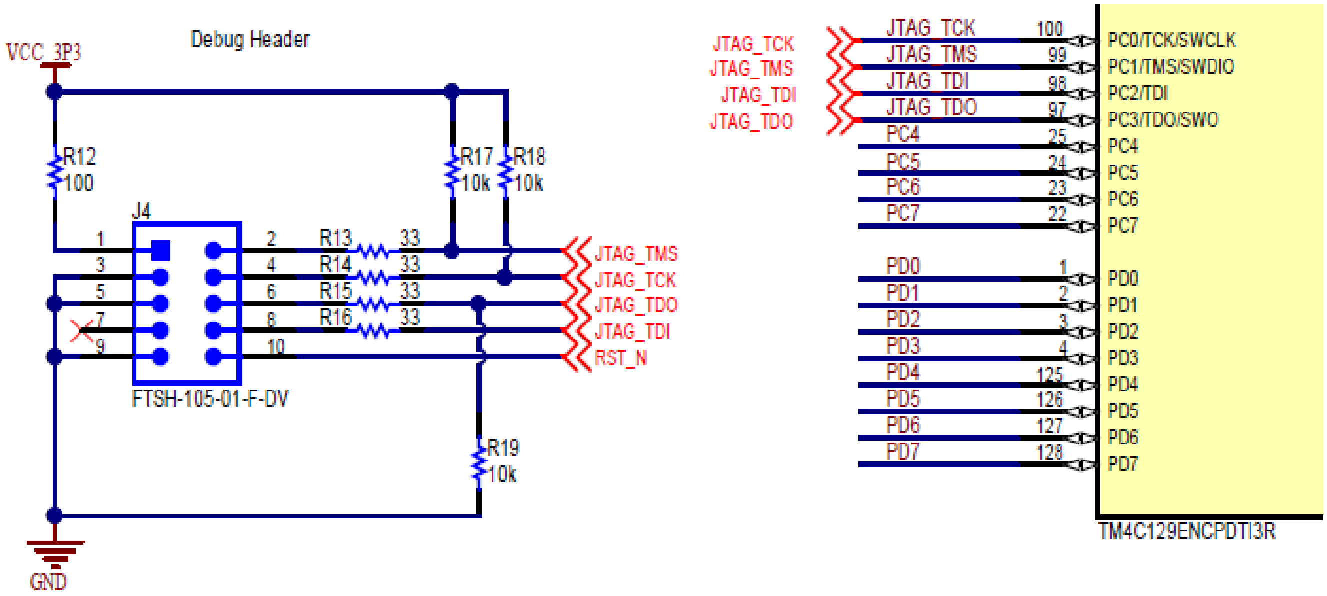


**Figure 7. ARM 10-Pin Header**

### 4.4 Sample Schematic for ARM 20-Pin Header

Figure 8 shows how the JTAG header on the board needs to be connected to the TM4C12x device for an ARM 20-pin header.

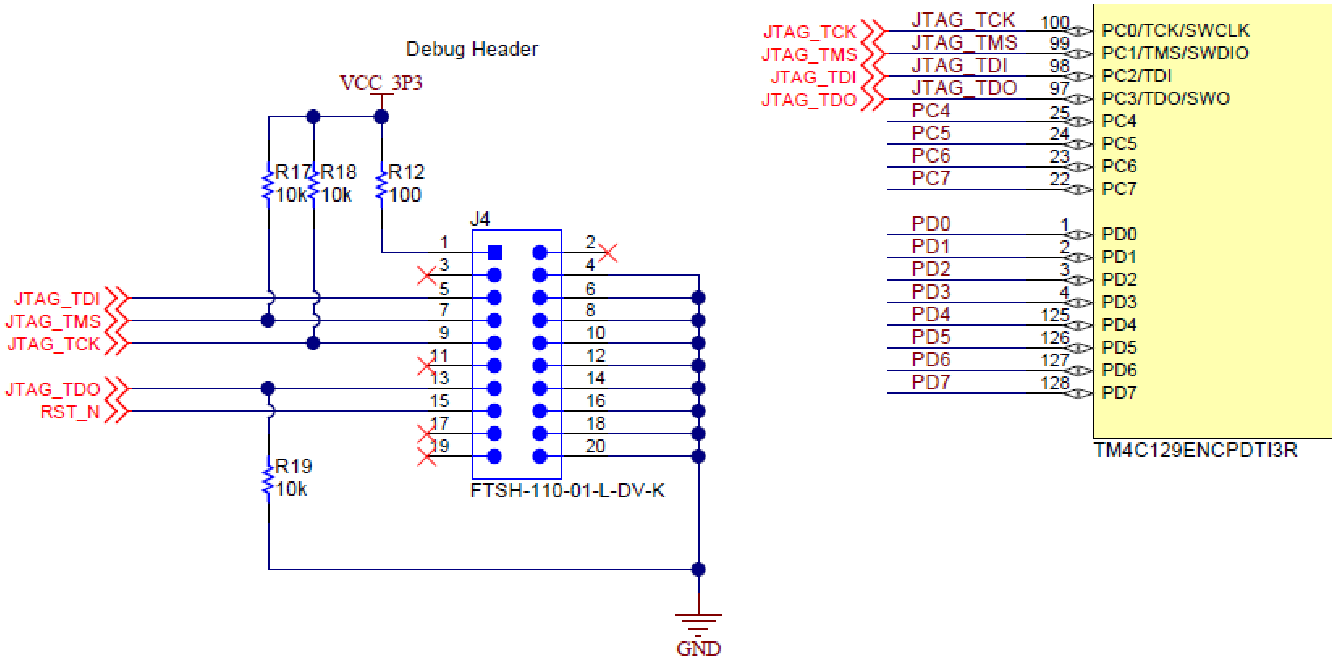


**Figure 8. ARM 20-Pin Header**

### 4.5 Using an External Standalone Debug Probe With EK-TM4C123GXL

The EK-TM4C123GXL LaunchPad comes with an on-board ICDI debug probe. However, you may want to use an external debug probe. The JTAG pins are available on the center of the board, but do not follow any of the standard JTAG header pin out. The following modifications should be performed on the LaunchPad to connect an external debug probe (see Figure 9).

- Install the headers for the JTAG pins, reset and EXTDBG.
- Move the power selector switch to the DEVICE side.
- Power to the board is now applied from the DEVICE USB connector. This is required because if the DEBUG USB connector is used, the on-board ICDI would also be active and may cause the PC to get confused on whether to use ICDI or external debug probe.

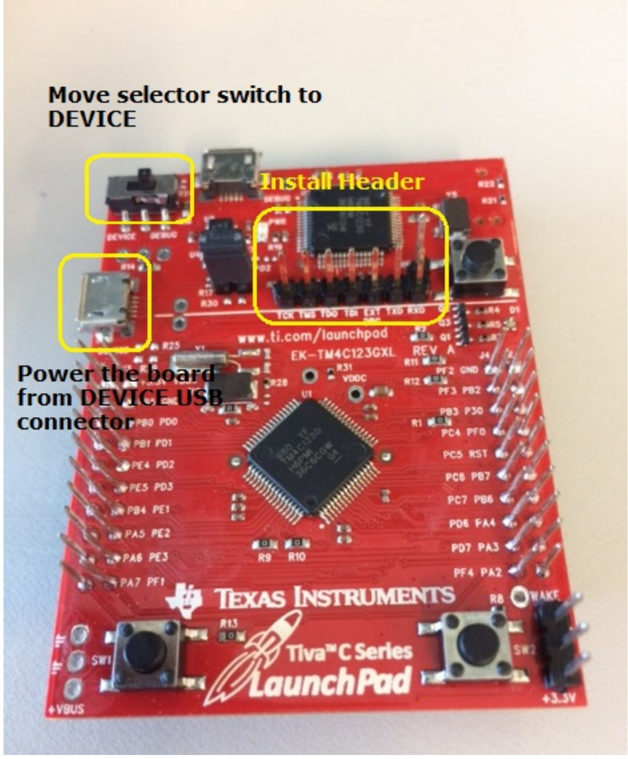

Move selector switch to DEVICE
Install Header
Power the board from DEVICE USB connector

**Figure 9. Preparing EK-TM4C123GXL for External Debug Probe**

Since there is no standard header available, the header needs to be wired to the external debug probe as shown in Figure 10. The following steps must be followed for correct operation.

1. Connect the header pin EXTDBG to GND on the EK-TM4C123GXL. This is required so that when the board is powered up, the ICDI detects a logic-low level and tri-state the JTAG pins allowing external debug probe to drive the signal.
2. Connect the header pin GND to GND on the debug probe
3. Connect the header pin TCK to TCK on the debug probe
4. Connect the header pin TMS to TMS on the debug probe
5. Connect the header pin TDI to TDI on the debug probe
6. Connect the header pin TDO to TDO on the debug probe
7. Connect the header pin RST_N to reset on the debug probe
8. Make sure that the power selector is moved to DEVICE side
9. Connect the USB cable on the USB device connector

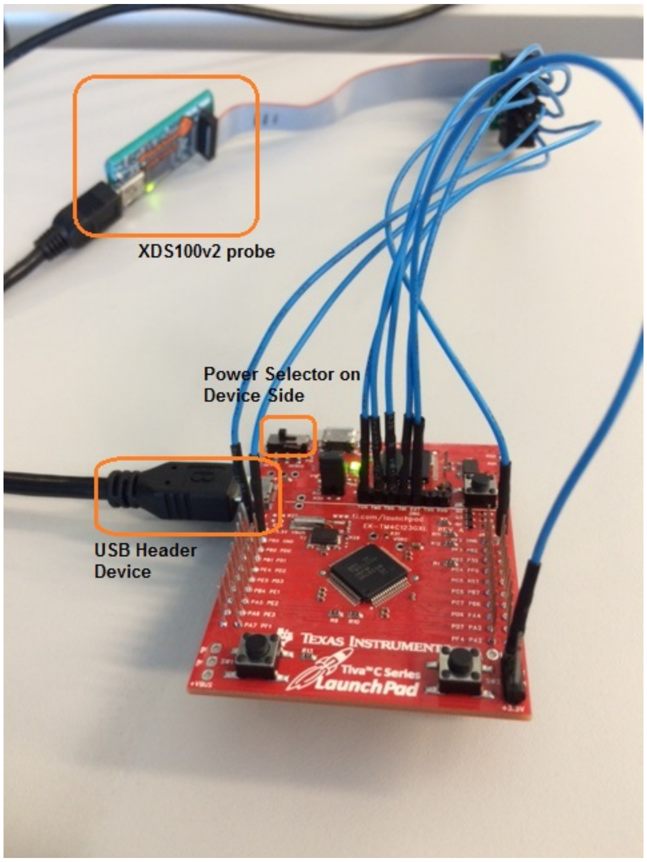


**Figure 10. Connecting External Debug Probe to EK-TM4C123GXL**

Now the external debug probe can be used to debug the main MCU on the EK-TM4C123GXL.

## 4.6 Using Stellaris ICDI on EK-TM4C123GXL to Debug Off-Board TM4C12x

The EK-TM4C123GXL on-board ICDI can also be used to debug an off-board TM4C12x. The following modifications should be performed on the LaunchPad to connect the ICDI on the LaunchPad as a debug probe to an external TM4C12x device (see Figure 11).

- Install the headers for the JTAG pins, reset and EXTDBG
- Keep the power selector switch to the DEBUG side
- Remove the 1x2 header between pins H24 and H25. This is required so that the main MCU is powered down and the TDO from the main MCU does not create an electrical bus contention with the off-board TM4C12x device's TDO.

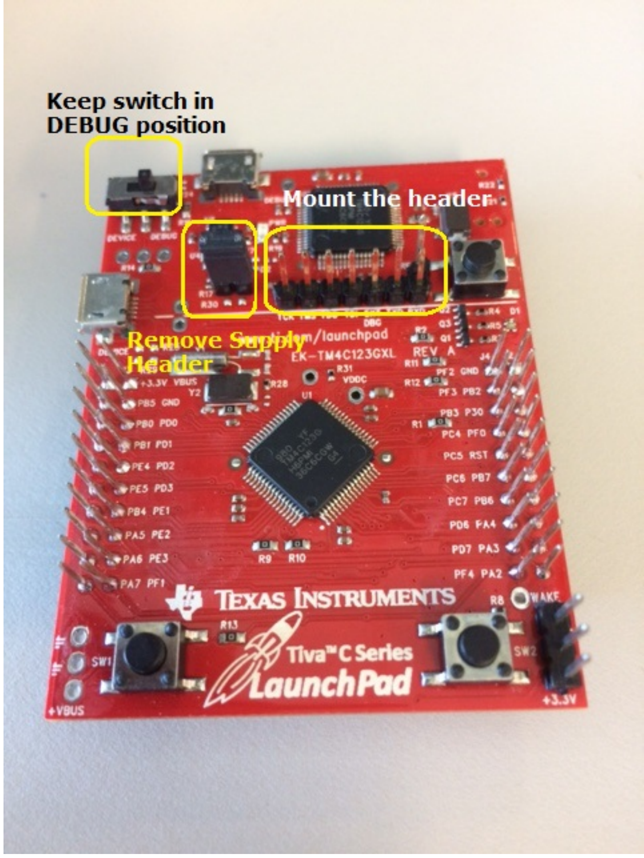


**Figure 11. Preparing EK-TM4C123GXL as a Debug Probe**

Since there is no standard header available, wire the header to the off-board TM4C12x (see Figure 12). The following steps must be followed for correct operation:

1. Ensure that header pin EXTDBG is unconnected
2. Connect the header pin GND from EK-TM4C123GXL to GND on the external board
3. Connect the header pin TCK from EK-TM4C123GXL to TCK on the off-board TM4C12x
4. Connect the header pin TMS from EK-TM4C123GXL to TNS on the off-board TM4C12x
5. Connect the header pin TDI from EK-TM4C123GXL to TDI on the off-board TM4C12x
6. Connect the header pin TDO from EK-TM4C123GXL to TDO on the off-board TM4C12x
7. Connect the header pin RST_N from EK-TM4C123GXL to RST_N on the off-board TM4C12x
8. Make sure that the power selector is moved to DEBUG side
9. Connect the USB cable on the USB debug connector

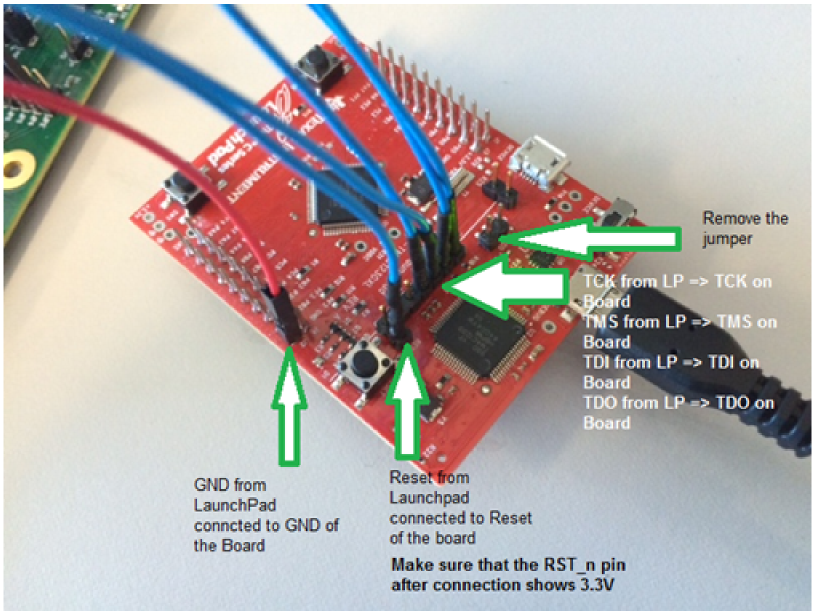

Remove the jumper
TCK from LP => TCK on Board
TMS from LP => TMS on Board
TDI from LP => TDI on Board
TDO from LP => TDO on Board
GND from LaunchPad conncted to GND of the Board
Reset from Launchpad connected to Reset of the board
Make sure that the RST_n pin after connection shows 3.3V

**Figure 12. Preparing EK-TM4C123GXL as a Debug Probe**

Now the ICDI can be used to debug an off-board TM4C12x device.

## 4.7 Using an External Standalone Debug Probe With EK-TM4C1294XL

The EK-TM4C1294XL LaunchPad comes with an on-board ICDI debug probe. However, you may want to use an external debug probe. The JTAG pins are available as an ARM 10-pin header marked U6 (see Figure 13). No board modifications are required when using an external standalone debug probe to connect to the main MCU on the EK-TM4C1294XL and the debug USB cable can still be used to provide power to the board.

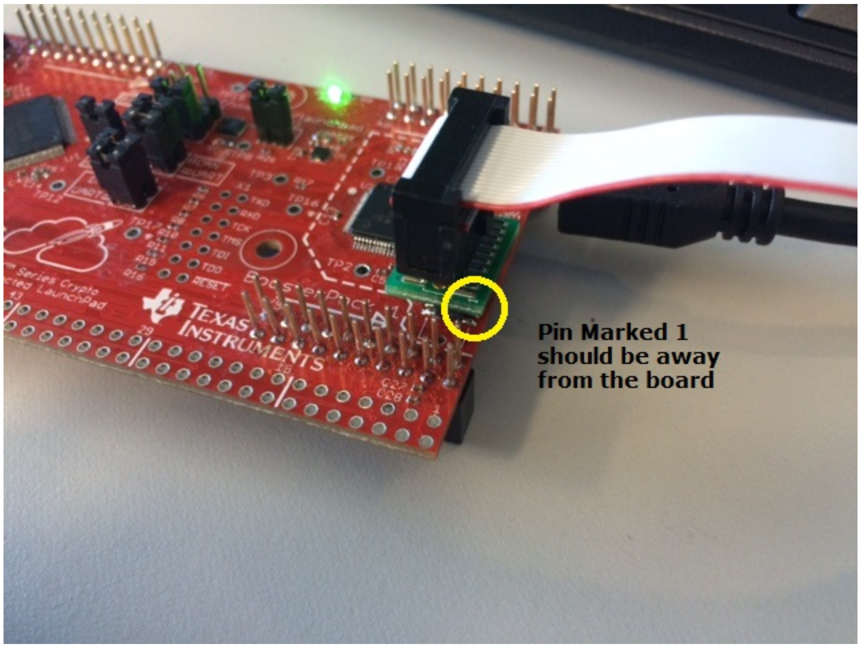


Figure 13. Connecting External Debug Probe to EK-TM4C1294XL

### 4.8 Using Stellaris ICDI on EK-TM4C1294XL to Debug Off-Board TM4C12x

The EK-TM4C1294XL on-board ICDI can also be used to debug an off-board TM4C12x. To be able to connect the on board ICDI as a debug probe, you must prepare the board (see Figure 14) as follows:

1. Remove the resistors R8, R10, R11, R15 and R16
2. Install the header X1 for the JTAG pins

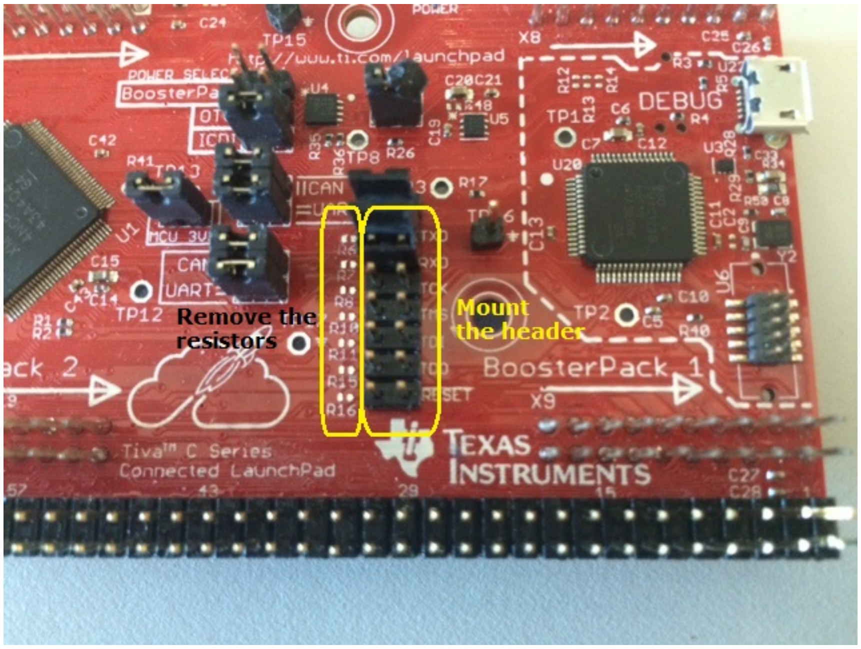


**Figure 14. Preparing EK-TM4C1294XL as a Debug Probe**

Since there is no standard header available, wire the header to the off-board TM4C12x (see Figure 15). The following steps must be followed for correct operation:

1. Connect the header pin GND from EK-TM4C1294XL to GND on the external board.
2. Connect the header pin TCK from EK-TM4C1294XL to TCK on the off-board TM4C12x.
3. Connect the header pin TMS from EK-TM4C1294XL to TNS on the off-board TM4C12x.
4. Connect the header pin TDI from EK-TM4C1294XL to TDI on the off-board TM4C12x.
5. Connect the header pin TDO from EK-TM4C1294XL to TDO on the off-board TM4C12x.
6. Connect the header pin RST_N from EK-TM4C1294XL to RST_N on the off-board TM4C12x.
7. Connect the USB cable on the debug connector.

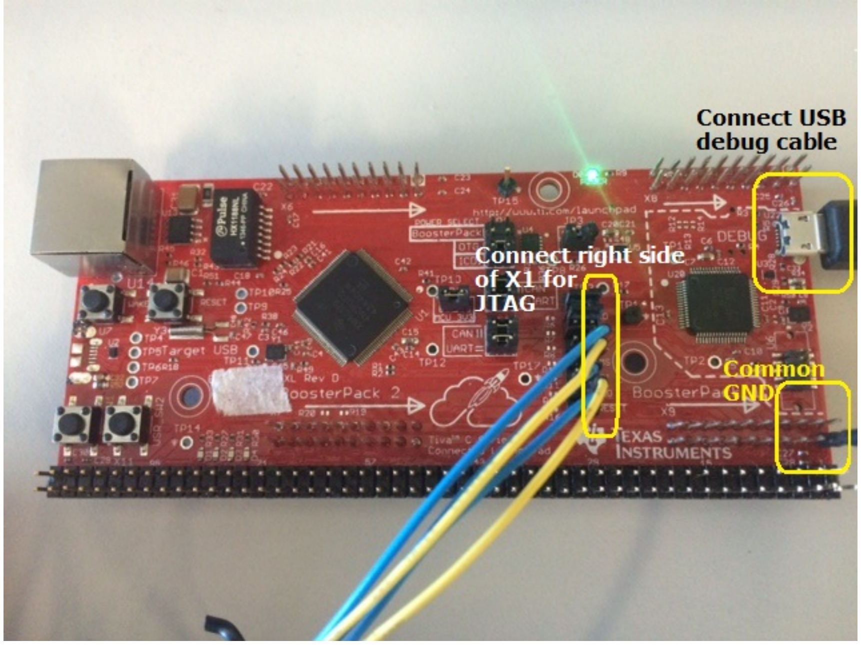


**Figure 15. Connecting EK-TM4C1294XL as a Debug Probe for Off-Board TM4C12x**

Now the on-board ICDI can be used to debug an off-board TM4C12x device.

### 4.9 JTAG Adapter

JTAG adapters are available from many JTAG debug probe vendors to convert one JTAG pinout to another. This is very useful because the correct JTAG probe may not have been procured or one is not readily available. Having the adapter allows you to use another JTAG debug probe with a different pinout to be interfaced. These adapters mostly come with a single conversion option. As part of this application report, the schematic (see Figure 16) shows a universal adapter that can be used for any of the 4 JTAG pinouts.

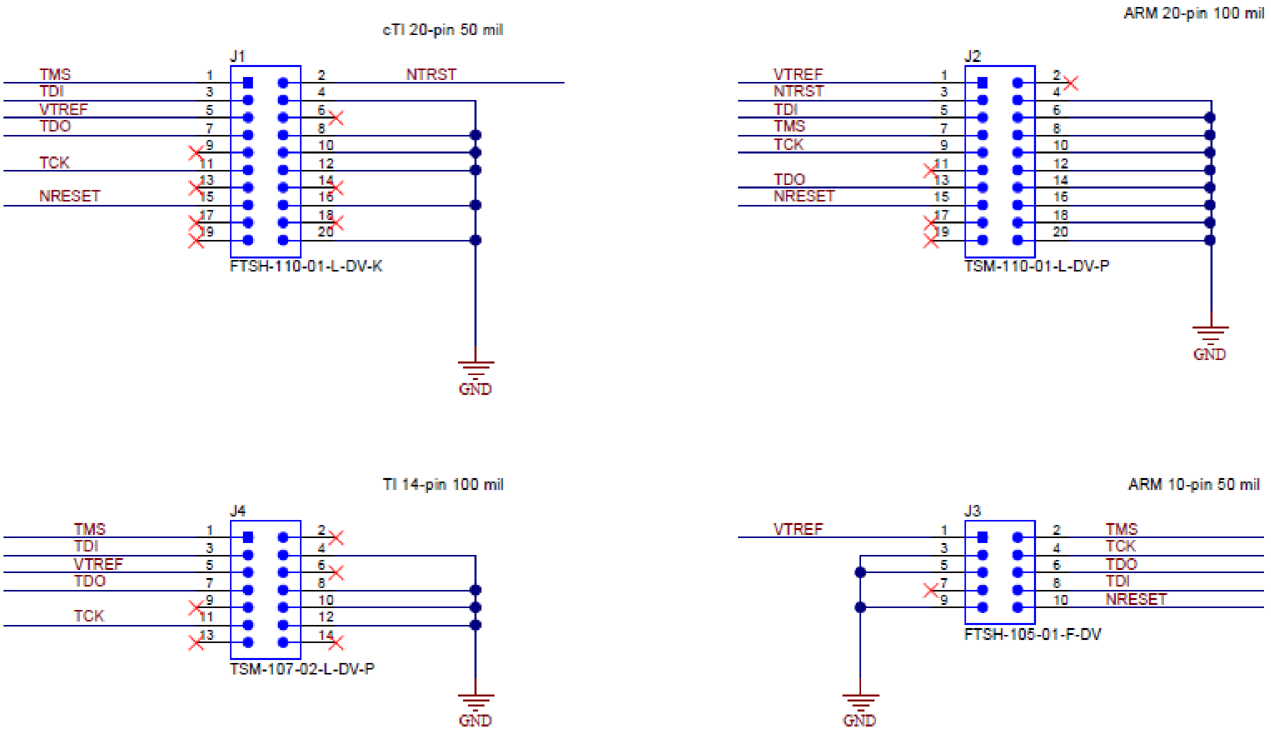


**Figure 16. JTAG Adapter Schematic**

## 5 TM4C12x JTAG Interface Specific Behaviors

This section describes the specific behavior of JTAG pins on the TM4C12x devices.

### 5.1 Default JTAG Pin Behavior on TM4C12x Devices

The pins PC0-3 after power on reset are configured as JTAG pins and not GPIO pins. Also, these pins are locked for JTAG operation by means of a commit control register. This provides a layer of protection against accidental programming of the pins to a non-JTAG function. If the customer application requires JTAG pins to be used as GPIO, then the application must perform the following steps:

1. Enable the Clock to GPIO port C, by setting bit-2 in system control RCGCGPIO register. The application must ensure that the system control register PRGPIO bit-2 reads 1 before accessing the GPIO port C address range.
2. Unlock the GPIO port C to access the commit control register by writing 0x4C4F434B to GPIOLOCK register of GPIO port C. A read of the register must return the value 0x0.
3. Set the commit control bits by writing bits 0-3 of the GPIOCR register of GPIO port C with the value 0xF.
4. Lock the GPIO Port C by writing any value other than 0x4C4F434B to the GPIOLOCK register of GPIO port C.

It is important to note that if the JTAG pins are configured as any other function, then the JTAG function will not be accessible on execution of the application. You must provide a mechanism in the application to either revert the GPIO to JTAG function, or hold the device in a while loop by reading the state of a GPIO pin before the GPIO is configured to be a non-JTAG function. If this is not done, the only method to access JTAG function is to execute the “Unlock Sequence” as provided in the device-specific data sheet.

### 5.2 Effect of BOOTCFG Register on JTAG Function

All TM4C12x devices have the BOOTCFG register. This register influences the behavior of the device boot after a power on reset. One of the functions is the ability to disable the JTAG function without configuring the GPIO port C as GPIO’s. This is achieved by clearing the bits 0 and 1. For more information, see device-specific data sheet. When the bits are cleared, the JTAG function is disabled.

If the register is not committed, a subsequent power on reset would configure it to its default value allowing for the JTAG function to work. However, if the register is committed, then a subsequent power on reset would make the change permanent. The only method to access the JTAG function is to execute the “Unlock Sequence” as provided in the device-specific data sheet.

### 5.3 Executing Unlock Sequence

The unlock sequence is a method to reset the TM4C12x microcontrollers to the factory state. When an unlock sequence is run, it has the following effect on the device.

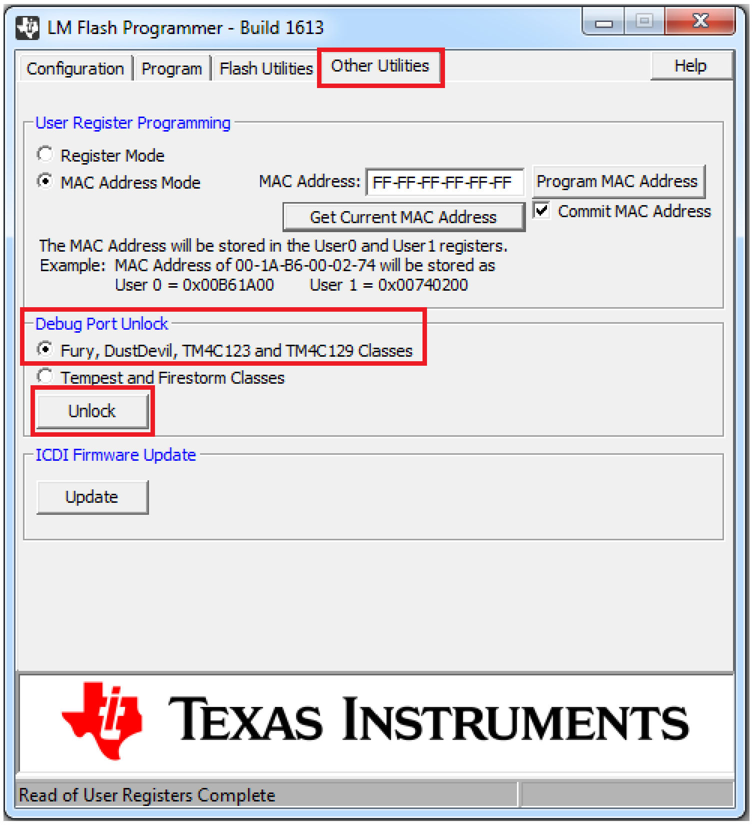
- Flash and EEPROM are erased
- All user committed registers are reset to the default value as per the device-specific data sheet

The actual protocol sequence is specified in the device datasheets in the “JTAG Interface” chapter. To execute the unlock sequence; the user must have a debug probe and software (see Table 4) that can perform this action.

### 5.3.1 Unlock Sequence using LMFlashProgrammer

The LMFlashProgrammer utility only supports the Stellaris ICDI for the unlock sequence. To execute the unlock sequence, follow the steps provided below:

1. Start the LMFlashProgrammer utility and go to the tab “Other Utilities” (see Figure 17). Select the radio button in front of “Fury, DustDevil, TM4C123 and TM4C129 Classes”.


LM Flash Programmer - Build 1613

User Register Programming

○ Register Mode
● MAC Address Mode

MAC Address: FF-FF-FF-FF-FF-FF

☑ Commit MAC Address

The MAC Address will be stored in the User0 and User1 registers.
Example: MAC Address of 00-1A-B6-00-02-74 will be stored as
User 0 = 0x00B61A00 User 1 = 0x00740200

Debug Port Unlock

● Fury, DustDevil, TM4C123 and TM4C129 Classes
○ Tempest and Firestorm Classes

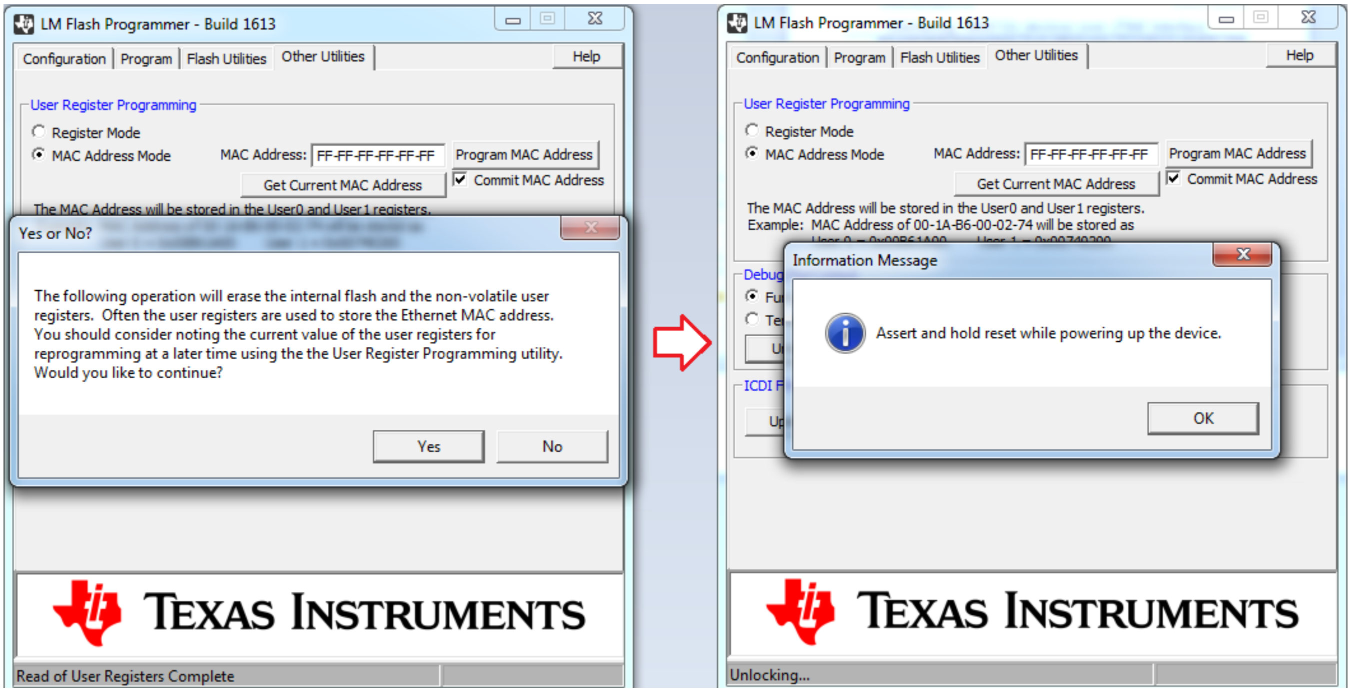
ICDI Firmware Update

Read of User Registers Complete

**Figure 17. LMFlashProgrammer Unlock Sequence Tab**

2. Power down the board containing the locked microcontroller.
3. Power up the board, while holding down the reset. The reset must remain pressed unless instructed by the LMFlashProgrammer to be released.

4. Press the “Unlock” button in LMFlashProgrammer. Press “Yes” in the pop up message box and then press “OK” in the information message box (see Figure 18).



User Register Programming

○ Register Mode

● MAC Address Mode

MAC Address: FF-FF-FF-FF-FF-FF

☑ Commit MAC Address

The MAC Address will be stored in the User0 and User1 registers.
Example: MAC Address of 00-1A-B6-00-02-74 will be stored as

Yes or No?

The following operation will erase the internal flash and the non-volatile user registers. Often the user registers are used to store the Ethernet MAC address. You should consider noting the current value of the user registers for reprogramming at a later time using the the User Register Programming utility. Would you like to continue?

Read of User Registers Complete

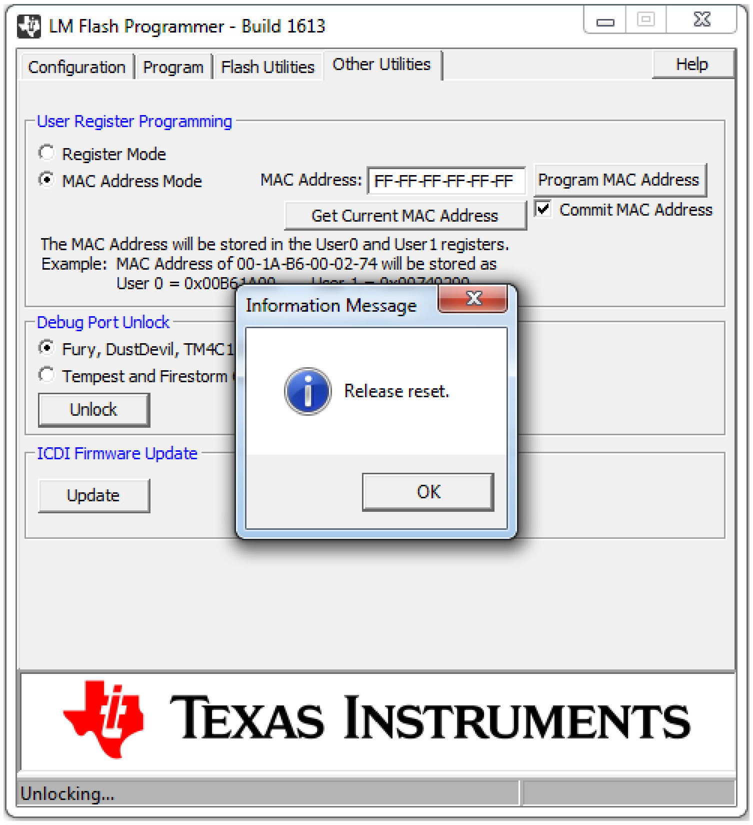
Information Message

Assert and hold reset while powering up the device.

Unlocking...

**Figure 18. Executing Unlock Sequence-1**

5. After some time, a pop up Information box will ask to release the reset. Release the reset and press OK in LMFlashProgrammer utility (see Figure 19).



○ Register Mode

● MAC Address Mode   MAC Address: FF-FF-FF-FF-FF-FF

☑ Commit MAC Address

The MAC Address will be stored in the User0 and User1 registers.
Example: MAC Address of 00-1A-B6-00-02-74 will be stored as
User 0 = 0x00B61A00   User 1 = 0x00740200

● Fury, DustDevil, TM4C1

○ Tempest and Firestorm

Release reset.

Unlocking...

**Figure 19. Executing Unlock Sequence-2**

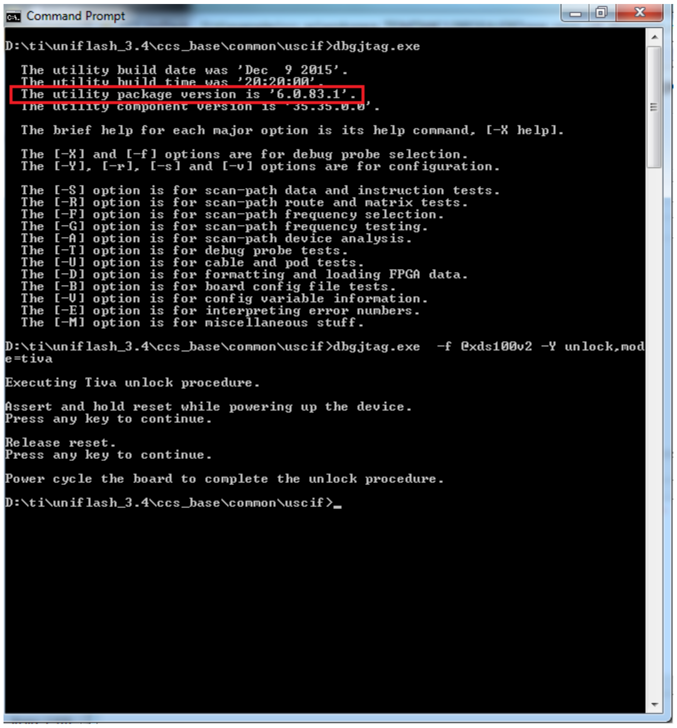
6. Finally, the board must be power cycled.

At the end of this step, the TM4C12x microcontroller should return to its factory state.

### 5.3.2 Unlock Sequence Using Uniflash and XDS Debug Probes

The Uniflash utility supports XDS100v2 and XDS200 debug probes if the utility package is “6.0.83.0” or latest. To execute the unlock sequence, follow the steps provided below:

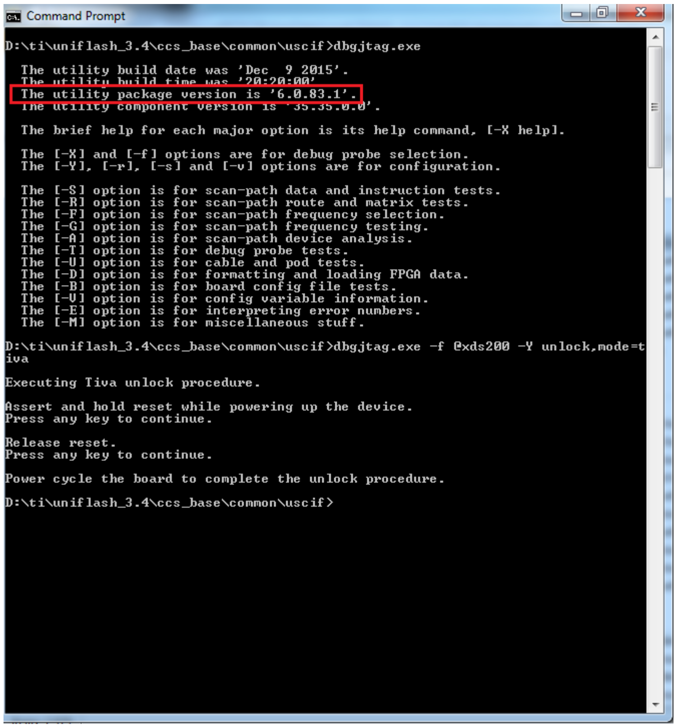
1. Start a windows command prompt and change the working directory to the path where dbgjtag.exe is present under Uniflash. For illustration, the path is “D:\ti\uniflash_3.4\ccs_base\common\uscif” for dbgjtag.exe.
2. Tun dbgjtag.exe to make sure the version of the utility package is at least “6.0.83.0” (see Figure 20 and Figure 21).
3. Type “dbgjtag.exe –f @xds100v2 –Y unlock,mode=tiva” if you are using XDS100v2 (see Figure 20).



```
D:\ti\uniflash_3.4\ccs_base\common\uscif>dbgjtag.exe

  The utility build date was 'Dec  9 2015'.
  The utility build time was '20:20:00'.
  The utility package version is '6.0.83.1'.
  The utility component version is '35.35.0.0'.

  The brief help for each major option is its help command, [-X help].

  The [-X] and [-f] options are for debug probe selection.
  The [-Y], [-r], [-s] and [-v] options are for configuration.

  The [-S] option is for scan-path data and instruction tests.
  The [-R] option is for scan-path route and matrix tests.
  The [-F] option is for scan-path frequency selection.
  The [-G] option is for scan-path frequency testing.
  The [-A] option is for scan-path device analysis.
  The [-T] option is for debug probe tests.
  The [-U] option is for cable and pod tests.
  The [-D] option is for formatting and loading FPGA data.
  The [-B] option is for board config file tests.
  The [-V] option is for config variable information.
  The [-E] option is for interpreting error numbers.
  The [-M] option is for miscellaneous stuff.

D:\ti\uniflash_3.4\ccs_base\common\uscif>dbgjtag.exe  -f @xds100v2 -Y unlock,mod
e=tiva

Executing Tiva unlock procedure.

Assert and hold reset while powering up the device.
Press any key to continue.

Release reset.
Press any key to continue.

Power cycle the board to complete the unlock procedure.

D:\ti\uniflash_3.4\ccs_base\common\uscif>_
```

**Figure 20. Unlock Sequence With Uniflash and XDS100v2**

4. Type “dbgjtag.exe –f @xds200 –Y unlock,mode=tiva” if you are using XDS200 (see Figure 21).



```
D:\ti\uniflash_3.4\ccs_base\common\uscif>dbgjtag.exe

  The utility build date was 'Dec  9 2015'.
  The utility build time was '20:20:00'
  The utility package version is '6.0.83.1'.
  The utility component version is '35.35.0.0'.

  The brief help for each major option is its help command, [-X help].

  The [-X] and [-f] options are for debug probe selection.
  The [-Y], [-r], [-s] and [-v] options are for configuration.

  The [-S] option is for scan-path data and instruction tests.
  The [-R] option is for scan-path route and matrix tests.
  The [-F] option is for scan-path frequency selection.
  The [-G] option is for scan-path frequency testing.
  The [-A] option is for scan-path device analysis.
  The [-T] option is for debug probe tests.
  The [-U] option is for cable and pod tests.
  The [-D] option is for formatting and loading FPGA data.
  The [-B] option is for board config file tests.
  The [-V] option is for config variable information.
  The [-E] option is for interpreting error numbers.
  The [-M] option is for miscellaneous stuff.

D:\ti\uniflash_3.4\ccs_base\common\uscif>dbgjtag.exe -f @xds200 -Y unlock,mode=t
iva

Executing Tiva unlock procedure.

Assert and hold reset while powering up the device.
Press any key to continue.

Release reset.
Press any key to continue.

Power cycle the board to complete the unlock procedure.

D:\ti\uniflash_3.4\ccs_base\common\uscif>
```

**Figure 21. Unlock sequence With Uniflash and XDS200**

5. Power down the board containing the locked microcontroller
6. Power up the board, while holding down the reset. The reset must remain pressed unless instructed by the Uniflash to be released.
7. Now press enter to execute the dbgjtag.exe with the options provided earlier and then follow the instructions given by dbgjtag.exe.
8. When dbgjtag.exe instructs to release the reset, then release the reset and power cycle the board.

At the end of this step, the TM4C12x microcontroller should return to its factory state.

# 6 Debugging JTAG Connection Failure

The JTAG is a very useful interface to debug embedded software. However, when the JTAG interface does not work as expected, it is extremely important to understand how to debug when JTAG is not functional.

## *6.1 JTAG Not Working on the First Bring Up*

When a new custom board is made, bringing up JTAG is the most critical part as only then can the firmware be programmed to the microcontroller. At the same time this is the most difficult debug for JTAG. Make sure that you have an XDS class debug probe, as it provides a utility called “Test Connection” (see Figure 22) that is useful in debugging the root cause.

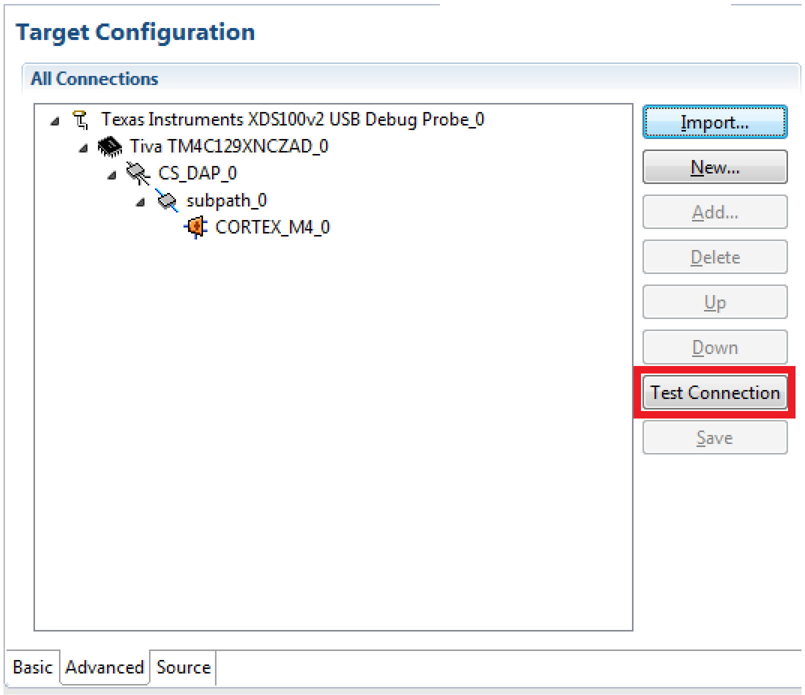
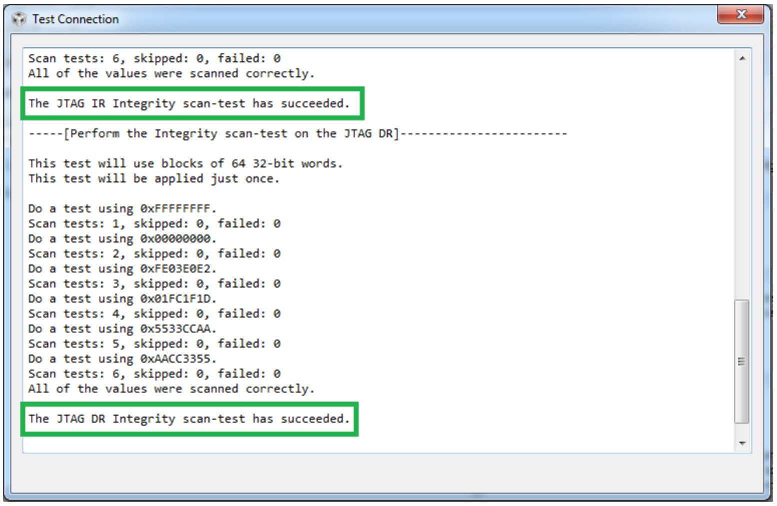


**Figure 22. Test Connection Utility in CCS**

When the button “Test Connection” is pressed, Code Composer Studio™ executes the JTAG instruction of BYPASS that checks whether the JTAG physical connection integrity can be established. The result is shown in a window (see Figure 23).



```
Scan tests: 6, skipped: 0, failed: 0
All of the values were scanned correctly.

The JTAG IR Integrity scan-test has succeeded.

-----[Perform the Integrity scan-test on the JTAG DR]------------------------

This test will use blocks of 64 32-bit words.
This test will be applied just once.

Do a test using 0xFFFFFFFF.
Scan tests: 1, skipped: 0, failed: 0
Do a test using 0x00000000.
Scan tests: 2, skipped: 0, failed: 0
Do a test using 0xFE03E0E2.
Scan tests: 3, skipped: 0, failed: 0
Do a test using 0x01FC1F1D.
Scan tests: 4, skipped: 0, failed: 0
Do a test using 0x5533CCAA.
Scan tests: 5, skipped: 0, failed: 0
Do a test using 0xAACC3355.
Scan tests: 6, skipped: 0, failed: 0
All of the values were scanned correctly.

The JTAG DR Integrity scan-test has succeeded.
```

**Figure 23. Test Connection Result**

If the JTAG IR and DR Integrity scan-test succeeds, it means that the device core is out of reset and may not have initialized itself. If, however, the integrity scan-test fails, the issue is in the power up process.

The following steps must be followed to make sure every known cause is eliminated till the source of the issue is found:

1. Check with a digital multi-meter that the VDD and VDDA supply rails are 3.3 V.
   (a) If not, then check the power supply aspect of the design.
2. Check with a digital multi-meter that the VDDC rail is 1.2 V.
   (a) If not and providing a power from an external power source, make sure that the current limit is set around 150 mA.
3. If the VDDC rail is at 1.2 V, make sure that the capacitance on the rail is as per the device-specific data sheet and the layout of the capacitors are as per the system design guidelines and application reports (see Section 8).
4. Check whether the JTAG header is correctly mounted and the TDIS pin (if available) is connected to GND.
5. Check on the JTAG header the VTREF pin is 3.3 V.
6. Check whether the Reset Pin of the microcontroller is at 3.3 V.
   (a) If not, connect an external pull up.
7. If using an external crystal oscillator, connect an oscilloscope on pin OSC0.
   (a) If the crystal is not oscillating, check the solder on the crystal and capacitors.
   (b) Always use the recommended crystals as per the device-specific data sheet.
8. If not using an external crystal oscillator, make sure that the pin OSC0 is connected to GND.
9. If using a TM4C129x device with integrated PHY, make sure that the RBIAS resistor is populated as per device-specific data sheet recommendation.

### 6.2 JTAG Not Working After Software Was Loaded

If software was downloaded to the TM4C12x and then it stopped working, the possible causes are limited and must be one of the items listed in Table 4.

**Table 4. Debugging JTAG Failure on Software Download**

| Possible Causes | Remedial Measure(s) |
|---|---|
| BOOTCFG was modified incorrectly | Execute the unlock sequence |
| GPIO Port C pins got modified to be a non-JTAG function | Execute the unlock sequence |
| System clock must be 10x the JTAG clock | If debug probe allows reducing JTAG clock then reduce JTAG clock frequency and re-try JTAG connection. If not then execute unlock sequence. |
| Device is in low power state | Wake up the device from low-power state and then re-try the JTAG connection. If it does not work, then execute the unlock sequence. |
| Power failure during software execution | There is a known erratum if the device is a TM4C123 revision 6. If it is applicable, replace the part with a revision 7. If not, execute the unlock sequence. |

## 7 Conclusion

This application report provides details on the JTAG interface of TM4C12x microcontrollers, the debug probes and the software tools that can be used to erase, program, debug and unlock the microcontroller. Also this document provides sample schematics for connecting different JTAG header pinouts with the TM4C12x microcontroller and an in-depth description on using TM4C12x LaunchPads as a debug probe. Finally, this document provides a detailed explanation on how to debug a JTAG connection failure, which is critical when bringing up an application-specific embedded system design.

## 8 References

The following related documents and software are available on the TM4C web page:

- Tiva™ C Series TM4C1294 Connected LaunchPad Evaluation Kit User's Guide
- Tiva™ C Series TM4C123G LaunchPad Evaluation Board User's Guide
- System Design Guidelines for the TM4C123x Family of Tiva™ C Series Microcontrollers
- System Design Guidelines for the TM4C129x Family of Tiva™ C Series Microcontrollers
- Processors wiki for JTAG
- XDS100 wiki page
- XDS110 wiki page
- XDS200 wiki page
- XDS560 wiki page

# 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