USB TAP Probe User Guide

Document Number: UTAPUG
Rev. 10.x, 06/2014

Contents

Section number **Title** **Page**

Chapter 1
Introducing CodeWarrior USB TAP probe

Chapter 2
Connecting to target and your host computer

Chapter 3
Using USB TAP probe

Chapter 1
Introducing CodeWarrior USB TAP probe

The CodeWarrior USB TAP probe is a cost-effective tool that helps you develop and debug a number of processors and microcontrollers. This chapter introduces you to the USB TAP probe.

This chapter explains:

CAUTION

The USB TAP probe contains components that are subject to damage from electrostatic discharge. Whenever you are using, handling, or transporting the USB TAP probe, or connecting to or disconnecting from a target system, always use proper anti-static protection measures, including static-free bench pads and grounded wrist straps.

1.1 What is USB TAP probe?

The USB TAP probe uses advanced emulation technology to provide control of and visibility into your target system. Combined with the CodeWarrior IDE, the USB TAP speeds the debugging process by letting you interactively control and examine the state of your target system.

Figure 1-1. USB TAP probe with USB 2.0 cable

1.1.1 Product highlights

The USB TAP probe has these features:

- Supports the following systems: Power Architecture processors, StarCore processors, 56800 Hybrid Controllers (processors and microcontrollers), and ColdFire™ processors.
- Supports all CPU core speeds.
- Lets you control and debug software running in-target, with minimal intrusion into target operation.
- Lets you debug code in cache, ROM, RAM, and flash memory.
- Provides high performance:
  - Split-second single-step execution.
  - Capable of download speeds greater than 12 MB per minute from host to target.

NOTE

The actual download speed depends on the target processor, the debug port's clock frequency, the network speed, and the debugger.

- Supports one USB 2.0 connection.
- Supports both big and little endian byte-order.
- Automatically supports target signal levels from 1.8V to 3.3V.
- Software debug capabilities including:
  - Controlling instruction execution.
  - Display and modify target memory.
  - Handles multicore and single core processors equally well.

- Examine and modify any processor registers.
- Run to breakpoints in ROM, RAM, or flash memory.
- Single-step through source and assembly language code views.
- Single-step into, over, or out of functions.

1.1.2 Debugging environment

The USB TAP probe works with the CodeWarrior debugger to give you control over the emulation functions and your target system.

1.1.3 USB TAP probe benefits

The USB TAP probe provides these key benefits:

- *Visibility*: The USB TAP probe makes it possible for you to observe registers and the current state of target memory. You can halt program execution at predefined states and examine the data for a particular program state.
- *Control*: You can conveniently control the state of the target system by downloading code, manually modifying processor registers and memory, single-stepping through the code, or setting breakpoints.

1.1.4 Target connections

The USB TAP probe connects to your target through the standard debug port for the processor family and supports a single target connection. The USB TAP probes are available in the following Freescale versions:

- JTAG/COP for Power Architecture™ targets
- DPI for Power Architecture™ MPC8xx and MPC5xx targets
- BDM for ColdFire® targets
- OnCE for StarCore, and 56800 Hybrid Controllers

For information on connecting to a target, see Connecting to target and your host computer topic.

1.2 Operating requirements

Before setting up your system, you should make sure that the operating environment is prepared.

1.2.1 Standard electrostatic precautions

This instrument contains static-sensitive components that are subject to damage from electrostatic discharge. Use standard ESD precautions when transporting, handling, or using the instrument and the target, when connecting/disconnecting the instrument and the target, and when removing the cover of the instrument.

We recommend that you use the following precautions:

- Use wrist straps or heel bands with a 1 MW resistor connected to ground.
- On the work surface and floor, use static conductive mats with a 1 MW resistor connected to ground.
- Keep high static-producing items, such as non-ESD-approved plastics, tape and packaging foam, away from the instrument and the target.

The above precautions should be considered as minimum requirements for a static-controlled environment.

1.2.2 Electrical requirements

The USB TAP probe is powered through the USB cable and does not use an external power supply. It is designed to be plugged directly into a host computer, but also can work with self-powered hubs. Bus-powered hubs may be unable to provide sufficient power for the USB TAP probe, which requires 200 mA. If insufficient power is available, your host operating system will indicate this failure and the probe will go into a low power suspend mode. If your hub is not able to provide sufficient power, connect the USB TAP probe directly to your host PC, or purchase a self-powered USB hub.

1.2.3 Operating temperature

The USB TAP probe can operate in a temperature range of 0 to 40 °C (32 to 104 ºF).

1.2.4 Target requirements

The USB TAP probe automatically supports target signal levels from 1.8V to 3.3V.

NOTE

In the case of the Power Architecture, for the USB TAP probe to properly stop and restart a JTAG/COP target processor, the $\overline{\text{QACK}}$ signal must be pulled low. The USB TAP probe pulls this signal low through the JTAG/COP connector.

1.3 Related documentation

This manual describes the procedures for unpacking the USB TAP probe, setting up USB communications, and connecting the probe to your target system.

The CodeWarrior documentation explains how to install and configure the CodeWarrior IDE and debugger.

Chapter 2
Connecting to target and your host computer

To run your software using the USB TAP probe, you must have working target hardware. This chapter explains how to connect the USB TAP probe to such hardware and to your host computer.

This chapter explains:

- Debug port connector information
- Connecting to target system
- Connecting to host computer
- Setting debug port clock frequency
- What to do next

CAUTION

The USB TAP probe contains components that are subject to damage from electrostatic discharge. Whenever you are using, handling, or transporting the USB TAP probe, or connecting to or disconnecting from a target system, always use proper anti-static protection measures, including using static-free bench pads and grounded wrist straps.

2.1 Debug port connector information

The USB TAP probe is a powerful development tool for use with a wide variety of processors that use either JTAG/COP, DPI, ColdFire BDM, or OnCE debug interfaces.

The following sections describe the debug port connector specifications:

- JTAG/COP connector information
- DPI connector information
- ColdFire BDM connector information
- OnCE connector information

2.2 Connecting to target system

CAUTION

Failure to properly connect the USB TAP probe to the target may damage the probe or target. Verify all connections before applying power.

The target system must have a debug port header that you can directly connect to the USB TAP probe. Make sure that you properly align the USB TAP multi-pin socket connector with the multi-pin header on your target system.

NOTE

Pin 1 is clearly marked on the gray ribbon cable by a red line down one side of the cable and a small triangle in the plastic socket.

2.2.1 Connecting USB TAP cable to target debug port header

1. Turn off the power to the target system.
2. Make sure that the USB cable from the USB TAP probe is not connected to the host computer.
3. Make sure that pin 1 of the gray ribbon cable connector aligns with pin 1 on the target's debug port header.

   The figure below shows pin 1 of the USB TAP probe.

Figure 2-1. USB TAP probe

4. Gently (but firmly) press the connector onto the target system debug port header.

This is how you can connect the USB TAP cable to the target debug port.

2.3 Connecting to host computer

To connect to the host computer:

1. Connect one end of a USB 2.0 cable to a USB port on the host computer.
2. Connect the other end of the cable to the CodeWarrior USB TAP probe USB connector.
3. Apply power to the target system.

This is how you can connect to the host computer.

2.4 Setting debug port clock frequency

The debug port on the target is a synchronous interface clocked by the TCK or DSCK signal. The base frequency is set by the debugger and can be adjusted when you set up a project in the debugger preferences or configuration options.

NOTE

For directions on how to set the debug port clock frequency, please see the CodeWarrior documentation.

Some slow targets might not be able to operate at the initial default rate. Therefore, you may have to adjust the debug port clock rate.

NOTE

Because of variations in the design of target systems, it is not possible to guarantee that all systems can be operated at the maximum debug port clock rates. These variations include circuit impedances, trace lengths, and signal terminations. You may need to select a lower clock rate to get reliable operation.

2.5 What to do next

If you have not already done so, you can now install the CodeWarrior software. See the IDE Users Guide or your targeting manual for information on how to configure the debugger and run a confidence test.

For additional information about using the USB TAP with your target system, see Using USB TAP probe topic.

Chapter 3
Using USB TAP probe

This chapter provides system startup procedures, explains how the USB TAP probe works, and provides important information about using the system. This chapter explains:

- USB TAP probe system startup
- Notes on using USB TAP probe

3.1 USB TAP probe system startup

This section explains how to start using the USB TAP probe and the debugger.

Before starting the USB TAP probe, make sure you have:

- Connected the USB TAP probe to your host computer with your USB 2.0 Cable.
- Connected the USB TAP probe to the target system (see Connecting to target and your host computer topic).
- Installed the debugger software and properly configured it to communicate with the USB TAP probe.

3.1.1 Starting USB TAP probe

1. Apply power to the target system.
2. Start the debugger.

   LEDs are provided to indicate the status of the USB TAP probe. For description of the various indicators, see Hardware specifications topic.

You are now ready to begin your debug session. For information on using the CodeWarrior debugger, see Targeting manual..

3.2 Notes on using USB TAP probe

The following topics provide information specific to USB TAP probe operation:

- Run/Pause/Mixed mode states
- Breakpoints in exception/interrupt handlers
- Connecting to multiple USB TAP probes

Also refer to the debugger documentation to become familiar with the system operation.

3.2.1 Run/Pause/Mixed mode states

When the host debugger is connected to the target via the USB TAP probe, the probe is always in one of these states (modes): *run,pause* or *mixed mode*. The Run/Pause LED on the probe will indicate the mode.

- Run mode - in this mode, all target system processor cores execute the target code. The Run/Pause LED will be green.
- Pause mode - in this mode, all target system processor cores have stopped executing the target code. The Run/Pause LED will be red.
- Mixed mode - in this mode, some target system processor cores are in run mode and others are in pause mode. The Run/Pause LED will be orange.

3.2.2 Breakpoints in exception/interrupt handlers

Care must be taken when setting breakpoints in exception handler code. A typical exception consists of a preamble that saves processor context, the actual exception handler, and then a postamble that restores processor context. You can use software breakpoints in the actual exception handler code, but not in the preamble or postamble where the processor context is changing.

NOTE

For Embedded Power Architecture processors, placing the CPU into debug mode is just another interrupt. For example: your code is in an interrupt epilogue and has just placed the return address into SRR0 when a breakpoint occurs. The breakpoint causes the IP for the address of the breakpoint to be written to SRR0, destroying your original return address. Stepping through code which accesses SRR0 and SRR1 exhibits the same problem.

To avoid this problem, always set your breakpoints before or after code which accesses SRR0 and SRR1, and never step through such code. For example, you can set your breakpoint anywhere after the interrupt prologue, but before the epilogue.

Instructions that involve the SRR0 and SRR1 registers are "MTSPR SRR0/1,Rx" "MFSPR Rx,SRR0/1," and "RFI."

3.2.3 Connecting to multiple USB TAP probes

You can connect to multiple USB TAP probes from one host computer in the CodeWarrior IDE, however, procedures may differ for each CodeWarrior IDE variant.

- For CodeWarrior tools that support creating multiple USB TAP probe connections in the IDE, simply define the connections, entering the unique probe serial number for each device. The IDE will manage the CCS sessions.
- For CodeWarrior tools that do not support creating multiple USB TAP probe connections in the IDE, create a CCS Remote Connection for each, using unique port numbers. Then for each device, start the CCS Console and configure the connection, specifying the probe serial number. Tools that support creation of only one USB TAP probe connection within the IDE will not provide an option for entering the device serial number.

Tip

If the CodeWarrior IDE variant requires using separate CCS sessions to connect to each USB TAP probe, and you would like the setup steps to run automatically when you launch the debugger, edit the `\ccs\bin\ccs.cfg` file with the new commands.

Tip

If you are using a USB hub to connect the USB TAP probe to the host computer, be sure to use a powered hub.

Tip

To set up the debug connection, you will need to know the 8-digit USB TAP probe serial number, located on a label on the bottom of the device.

Chapter 4
Hardware specifications

This chapter provides hardware specifications for the USB TAP probe.

This chapter explains:

- Connectors and LEDs
- USB TAP probe specifications

4.1 Connectors and LEDs

The following figures show the various LEDs and connectors of the USB TAP probe.

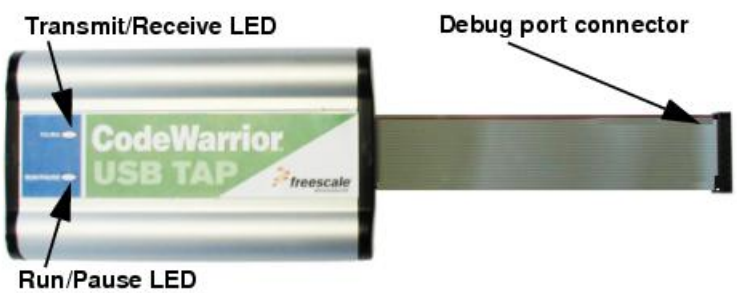

Figure 4-1. USB TAP probe - top view

Figure 4-2. USB TAP probe connector - end view

4.1.1 Run/Pause indicator

When the USB TAP probe is powered on, it performs a brief self-test that displays a test pattern on both the Run/Pause and Transmit/Receive LEDs. After the self-test completes, the status LED (labeled RUN/PAUSE) indicates the state of execution on the target device as follows:

- The LED is green when the target is running.
- The LED is red when the target is paused.
- The LED is orange when the target is in mixed mode.
- The LED is initially unlit and remains so until the debugger is connected to the USB TAP probe.

For the definition of run and pause modes, see Run/Pause/Mixed mode states topic.

4.1.2 Transmit/Receive indicator

When the USB TAP probe is powered on, it performs a brief self-test that displays a test pattern on both the Run/Pause and Transmit/Receive LEDs. After the self-test completes, the transmit/receive LED (labeled TX/RX) indicates the state of the USB interface as follows:

- The LED flashes red when the USB TAP probe is powered but has not been configured.
- The LED flashes green when the USB TAP probe is properly configured.
- The LED flashes orange when data is being transferred.
- The LED is unlit if the USB TAP probe is disabled or disconnected.

4.1.3 USB connector

The USB interface consists of a USB type B connector that connects directly to the USB cable provided.

4.1.4 Debug port connector

The debug port connector consists of a 5.5 inch ribbon cable with the appropriate debug connector attached. The ribbon cable has a red stripe down one side to indicate the location of pin 1.

NOTE

The OnCE connector is equipped with a removable plug in pin 8. This follows the keying convention for the OnCE header (pin 8 should be removed). This plug is removable, in case pin 8 is not removed from the target OnCE header.

4.2 USB TAP probe specifications

Below figure shows the dimensions of the USB TAP probe.

Figure 4-3. USB TAP probe dimensions

4.2.1 Electrical characteristics

The USB TAP probe affects the target processor and target electrical characteristics as little as possible. Care, however, should be taken in designing the target to accommodate the small signal delays associated with any in-circuit probe, emulator, or other test equipment.

The USB TAP probe automatically supports target signal levels from 1.8V to 3.3V.

4.2.2 Physical characteristics

The USB TAP probe was carefully designed to be as small as possible. Even so, it may not physically fit in all target systems.

Contact Freescale if you have any special considerations that need review.

Below table shows the physical characteristics of the USB TAP probe.

Table 4-1. USB TAP probe - physical characteristics

Physical characteristics	
Power consumption	
USB TAP current consumption from USB cable	< 200 mA
USB TAP current consumption from target	< 50 mA
Environmental Requirements	
Operating temperature	0 to 40 °C (32 to 104 °F)
Storage temperature	-40 to 70 °C (-40 to 158 °F)
Humidity	5% to 95% relative humidity,non-condensing
Physical	
USB TAP probe dimensions	5.875" x 3.000" x 1.250"(14.92 cm x 7.620 cm x 3.175 cm)
Target connector ribbon length	5.50" (13.97 cm)
Target connector dimensions	
Height (out of probe tip enclosure; above board)	0.38" (0.97 cm)
Thickness	0.20" (0.51 cm)
Pin-to-pin spacing	0.1" (0.25 cm)
JTAG/COP width	0.98" (2.49 cm)
DPI width	0.68" (1.72 cm)
OnCE width	0.88" (2.24 cm)
ColdFire BDM width	1.48" (3.76 cm)

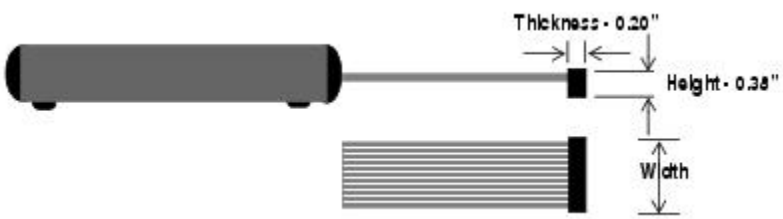

Figure 4-4. Target connector dimensions

Chapter 5
JTAG/COP connector information

The CodeWarrior USB TAP JTAG/COP probe has a 16-pin connector which automatically supports target signal levels from 1.8V to 3.3V.

The following figure shows the pin assignments of the probe JTAG/COP connector.

The following table lists JTAG/COP signal names, direction, pin numbers, descriptions, and drive capabilities for the probe JTAG/COP connector.

NOTE

All JTAG/COP signals must meet accepted standards for JTAG/COP signal design. To ensure proper and stable operation between the USB TAP probe and the target, the JTAG/COP signals must meet the requirements listed in Table A.2.

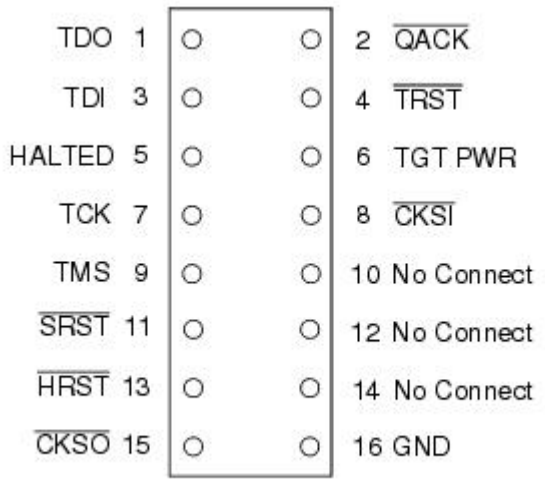

Figure 5-1. USB TAP probe for JTAG/COP connector pin assignments

Table 5-1. USB TAP probe for JTAG/COP signal directions

JTAG/COP pin	Signal mnemonic	Signal direction	Description
1	TDO	From target system	30pF load
2	QACK	From USB TAP probe connector	100Ohm pull-down
3	TDI	From USB TAP probe connector	50mA driver
4	TRST	From USB TAP probe connector	50mA driver
5	HALTED	Bi-directional	Open-drain, 100Ohm to ground when asserted by USB TAP probe, 35pF load when not asserted
6	TGT PWR	From target	2MOhm pull-down, plus 0.01uF load
7	TCK	From USB TAP probe connector	50mA driver
8	CKSI	From USB TAP probe connector	50mA driver
9	TMS	From USB TAP probe connector	50mA driver
10	No Connect	- n/a -	
11	SRST	Bi-directional	Open-drain. 100Ohm to ground when asserted by USB TAP, 35pF load when not asserted[1]
12	No Connect	- n/a -	
13	HRST	Bi-directional	Open-drain. 100Ohm to ground when asserted by USB TAP, 35pF load when not asserted[1]

Table continues on the next page...

Table 5-1. USB TAP probe for JTAG/COP signal directions (continued)

JTAG/COP pin	Signal mnemonic	Signal direction	Description
14	No Connect	- n/a -	
15	$\overline{\text{CKSO}}$	From target	30pF load[1]
16	GND	- n/a -	

1. 4.7KOhm pull-up to buffered TGT PWR.

The following table provides a general description of each JTAG/COP signal and the operational requirements.

Table 5-2. USB TAP probe for JTAG/COP signal recommendations/requirements

JTAG/COPpin	Signal mnemonic	Requirement
1	TDO	Must be wired to the target processor. TDO is an output from the target processor and an input to the USB TAP probe. The TDO trace run should be kept short and maintain a "two-signal-width" spacing from any other parallel dynamic signal trace. TDO should have a series termination resistor located near the target processor.
2	$\overline{\text{QACK}}$	May be wired to the target processor. $\overline{\text{QACK}}$ is an input to most Power Architecture processors and must remain low while the USB TAP probe is connected to the target. The USB TAP probe connects this signal internally to the JTAG/COP GND pin (16) through a 100Ohm resistor.
3	TDI	Must be wired to the target processor. The USB TAP probe drives the TDI output with up to 50mA. The TDI trace should be kept short and maintain a "two-signal-width" spacing from any other parallel dynamic signal trace. TDI should have an RC termination option at the processor.
4	$\overline{\text{TRST}}$	Must be wired to the target processor. The USB TAP probe drives the $\overline{\text{TRST}}$ output with up to 50mA. To gain control of the processor, the USB TAP probe negates $\overline{\text{TRST}}$ approximately 250 milliseconds before negation of $\overline{\text{HRST}}$. This allows the USB TAP probe to issue COP commands through the JTAG/COP interface and gain control of the processor upon negation of $\overline{\text{HRST}}$. The $\overline{\text{TRST}}$ trace run should be kept short and maintain a "two-signal-width" spacing from any other parallel dynamic signal trace.

Table continues on the next page...

Table 5-2. USB TAP probe for JTAG/COP signal recommendations/requirements (continued)

JTAG/COPpin	Signal mnemonic	Requirement
5	HALTED	Need not be wired to the target. The USB TAP probe does not currently use this signal.
6	TGT PWR	Must be wired to the target. The USB TAP probe uses this signal to determine if power is applied to the target. This signal is also used as a voltage reference for the signals driven by the USB TAP probe ($\overline{\text{CKSI}}$, $\overline{\text{TRST}}$, TCK, TMS, TDI). TGT PWR (pin 6) should be connected to target Vcc through a 1KOhm pull-up.
7	TCK	Must be wired to the target processor. The USB TAP probe drives the TCK output with up to 50mA. The TCK trace run should be kept as short as possible and maintain a "two-signal-width" spacing from any other parallel dynamic signal trace.
8	CKSI	Need not be wired to the target.The USB TAP does not currently use this signal.
9	TMS	Must be wired to the target processor. The USB TAP drives the TMS output with up to 50mA. TMS should be kept as short as possible and maintain a "two-signal-width" spacing from any other parallel dynamic signal trace. TMS should have a termination option at the processor.
10	No Connect	Not required for emulation.
11	$\overline{\text{SRST}}$	May be wired to the target processor. During reset, the USB TAP drives $\overline{\text{SRST}}$ to ground through a 100Ohm resistor.
12	No Connect	Not required for emulation.
13	$\overline{\text{HRST}}$	Must be wired to the target processor. During reset, the USB TAP probe drives $\overline{\text{HRST}}$ to ground through a 100Ohm resistor.
14	No Connect	Not required for emulation.
15	$\overline{\text{CKSO}}$	Should be wired to the target processor. The USB TAP senses $\overline{\text{CKSO}}$ to determine if the processor halted execution in a checkstop state.
16	GND	Must be wired to the target. GND is connected directly to the ground inside the USB TAP probe.

5.1 Notes specific to MPC8240, MPC8241, and MPC8245

$\overline{\text{CKSO}}$ is not available on the MPC8240, MPC8241, or MPC8245 processors.

$\overline{\text{SRST}}$ can be deselected in favor of the SDMA12 signal used for Extended Addressing (except on the MPC8240 processor).

$\overline{\text{QACK}}$ is an output signal only, so it need not be connected to the COP header.

The USB TAP probe fully supports the MPC8240, MPC8241, and MPC8245 processors despite the absence of the signals mentioned above.

5.1.1 Signal width example

Signal line 'A' is 0.005 mil. An adjacent dynamic signal line 'B' should maintain a "two-signal-width" distance from signal line 'A'. So that from the center of line 'A' to the center of line 'B', there should be 0.0025 + 0.005 + 0.005 + 0.0025 = 0.015mil.

Target bias should maintain "one-signal-width" spacing from the signal.

Chapter 6
DPI connector information

The CodeWarrior USB TAP DPI probe has a 10-pin connector which automatically supports target signal levels from 1.8V to 3.3V.

The following figure shows the pin assignments of the probe DPI connector.

The following table lists DPI signal names, direction, pin numbers, descriptions, and drive capabilities for the probe DPI Connector.

NOTE

All DPI signals must meet accepted standards for DPI signal design. To ensure proper and stable operation between the USB TAP probe and the target, the DPI signals must meet the requirements listed in Table B.2.

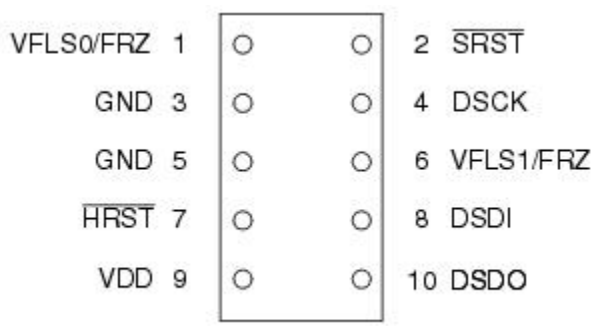

Figure 6-1. USB TAP probe for DPI connector pin assignments

Table 6-1. USB TAP probe for DPI signal directions

DPI pin	Signal mnemonic	Signal direction	Description
1	VLSO/FRZ	From target system	30pF load
2	SRST	Bi-directional	Open-drain. 100Ohm to ground when asserted by USB TAP probe, 35pF load when not asserted[1]
3	GND	- n/a -	

Table continues on the next page...

Table 6-1. USB TAP probe for DPI signal directions (continued)

DPI pin	Signal mnemonic	Signal direction	Description
4	DSCK	From USB TAP probe connector	50mA driver
5	GND	- n/a -	
6	VFLS1/FRZ	From target system	35pF load[1]
7	$\overline{\text{HRST}}$	Bi-directional	Open-drain. 100Ohm to ground when asserted by USB TAP probe, 35pF load when not asserted[1]
8	DSDI	From USB TAP probe connector	50mA driver
9	VDD	From target system	2MOhm pull-down, plus 0.01uF load
10	DSDO	From target system	30pF load

1. 4.7KOhm pull-up to buffered TGT PWR.

The following table provides a general description of each DPI signal and the operational requirements.

Table 6-2. USB TAP probe for DPI signal recommendations and requirements

DPI pin	Signal mnemonic	Requirement
1	VLSO/FRZ	VFLS0/FRZ is not needed for emulation.
2	$\overline{\text{SRST}}$	Must be wired to the target. During reset, the USB TAP probe drives $\overline{\text{SRST}}$ to ground through a 100Ohm resistor.
3	GND	Must be wired to the target. GND is connected directly to the ground inside the USB TAP probe.
4	DSCK	Must be wired to the target processor. It is driven by the USB TAP as an output with up to 50mA. This signal is the clock for the DPI interface. It is good design practice to keep the trace length short and isolate the trace from other signals. If the trace must be long, then termination may be needed.
5	GND	Must be wired to the target. GND is connected directly to the ground inside the USB TAP probe.
6	VFLS1/FRZ	VFLS1/FRZ is not needed for emulation.
7	$\overline{\text{HRST}}$	Must be wired to the target. During reset, the USB TAP probe drives $\overline{\text{HRST}}$ to ground through a 100Ohm resistor.
8	DSDI	Must be wired to the target processor. The USB TAP probe drives the TDI output with up to 50 mA.

Table continues on the next page...

Table 6-2. USB TAP probe for DPI signal recommendations and requirements (continued)

DPI pin	Signal mnemonic	Requirement
9	VDD	Must be wired to the target. The USB TAP probe uses this signal to determine if power is applied to the target. This signal is also used as a voltage reference for the signals driven by the USB TAP probe (SRST, SDCK, HRST, DSDI).
10	DSDO	Must be wired to the target processor. DSDO is an output from the target processor and an input to the USB TAP probe. It is good design practice to keep the trace length short and isolate the trace from the other signals.

Chapter 7
ColdFire BDM connector information

The CodeWarrior USB TAP ColdFire BDM probe has a 26-pin connector which automatically supports target signal levels from 1.8V to 3.3V.

The following figure shows the pin assignments of the probe BDM connector.

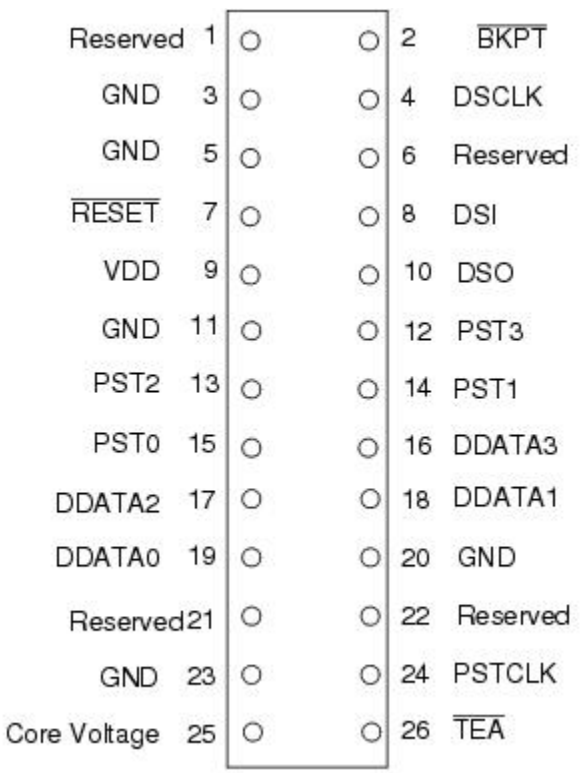

Figure 7-1. USB TAP probe for ColdFire BDM connector pin assignments

The table below lists BDM signal names, direction, pin numbers, descriptions, and drive capabilities for the probe BDM connector.

Table 7-1. USB TAP probe for ColdFire BDM signal directions

BDM pin	Signal mnemonic	Signal direction	Description
1	Reserved	From target system	30pF load

Table continues on the next page...

Table 7-1. USB TAP probe for ColdFire BDM signal directions (continued)

BDM pin	Signal mnemonic	Signal direction	Description
2	BKPT	Bi-directional	Open-drain. 100Ohm to ground when asserted by USB TAP probe, 35pF load when not asserted[1]
3	GND	- n/a -	
4	DSCLK	From USB TAP probe connector	50mA driver
5	GND	- n/a -	
6	Reserved	- n/a -	
7	$\overline{\text{RESET}}$	Bi-directional	Open-drain. 100Ohm to ground when asserted by USB TAP probe, 35pF load when not asserted[1]
8	DSI	From USB TAP probe connector	50mA driver
9	VDD	From target system	2MOhm pull-down, plus 0.01uF load
10	DSO	From target system	30pF load
11	GND	- n/a -	
12	PST3	- n/a -	
13	PST2	- n/a -	
14	PST1	- n/a -	
15	PST0	- n/a -	
16	DDATA3	- n/a -	
17	DDATA2	- n/a -	
18	DDATA1	- n/a -	
19	DDATA0	- n/a -	
20	GND	- n/a -	
21	Reserved	- n/a -	
22	Reserved	- n/a -	
23	GND	- n/a -	
24	PSTCLK	From target	30pF load
25	Core Voltage	- n/a -	
26	$\overline{\text{TEA}}$	- n/a -	

1. 4.7Ohm pull-up to buffered TGT PWR.

NOTE

All BDM signals must meet accepted standards for ColdFire BDM signal design. To ensure proper and stable operation between the USB TAP probe and the target, the BDM signals must meet the requirements listed in table below.

The following table provides a general description of each BDM signal and the operational requirements.

Table 7-2. USB TAP probe for ColdFire BDM signal recommendations and requirements

BDM pin	Signal mnemonic	Requirement
1	Reserved	Need not be wired to the target. The USB TAP probe does not currently use this signal.
2	BKPT	Must be wired to the target.
3	GND	Must be wired to the target. GND is connected directly to the ground inside the USB TAP probe.
4	DSCLK	DSCLK must be connected to the target's processor. It is driven by the USB TAP probe as an output with up to 50mA. This signal is the clock for the BDM interface. It is good design practice to keep the trace length short and isolate the trace from other signals. If the trace must be long, then termination may be needed.
5	GND	Must be wired to the target. GND is connected directly to the USB TAP ground inside the USB TAP probe.
6	Reserved	Need not be wired to the target. The USB TAP probe does not currently use this signal.
7	$\overline{\text{RESET}}$	Must be wired to the target. During reset, the USB TAP probe drives $\overline{\text{RESET}}$ to ground through a 100Ohm resistor.
8	DSI	Must be wired to the target processor. The USB TAP probe drives the TDI output with up to 50mA.
9	VDD	Must be wired to the target. The USB TAP probe uses this signal to determine if power is applied to the target. The signal is also used as a voltage reference for the signals driven by the USB TAP probe ($\overline{\text{BKPT}}$, DSCLK, $\overline{\text{RESET}}$, DSI, and $\overline{\text{TEA}}$).
10	DSO	Must be wired to the target processor. It is an input to the USB TAP probe.
11	GND	Must be wired to the target. GND is connected directly to the ground inside the USB TAP probe.
12	PST3	Need not be wired to the target. The USB TAP probe does not currently use this signal.
13	PST2	Need not be wired to the target. The USB TAP probe does not currently use this signal.

Table continues on the next page...

Table 7-2. USB TAP probe for ColdFire BDM signal recommendations and requirements (continued)

BDM pin	Signal mnemonic	Requirement
14	PST1	Need not be wired to the target. The USB TAP probe does not currently use this signal.
15	PST0	Need not be wired to the target. The USB TAP probe does not currently use this signal.
16	DDATA3	Need not be wired to the target. The USB TAP probe does not currently use this signal.
17	DDATA2	Need not be wired to the target. The USB TAP probe does not currently use this signal.
18	DDATA1	Need not be wired to the target. The USB TAP probe does not currently use this signal.
19	DDATA0	Need not be wired to the target. The USB TAP probe does not currently use this signal.
20	GND	Must be wired to the target. GND is connected directly to the ground inside the USB TAP probe.
21	Reserved	Need not be wired to the target. The USB TAP probe does not currently use this signal.
22	Reserved	Need not be wired to the target. The USB TAP probe does not currently use this signal.
23	GND	Must be wired to the target. GND is connected directly to the ground inside the USB TAP probe.
24	PSTCLK	May be wired to the target. The USB TAP probe uses this signal to support synchronous clocking mode.
25	Core Voltage	Need not be wired to the target. The USB TAP probe does not currently use this signal.
26	$\overline{\text{TEA}}$	Need not be wired to the target. The USB TAP probe does not currently use this signal.

Chapter 8
OnCE connector information

The CodeWarrior USB TAP OnCE probe has a 14-pin connector which automatically supports target signal levels from 1.8V to 3.3V.

The following figure shows the pin assignments of the probe OnCE connector.

Figure 8-1. USB TAP probe for OnCE connector pin assignments

The following table lists OnCE signal names, direction, pin numbers, descriptions, and drive capabilities for the probe OnCE Connector.

Table 8-1. USB TAP probe signal directions

OnCE pin	Signal mnemonic	Signal direction	Description
1	TDI	From USB TAP probe connector	50mA driver
2	GND	- n/a -	
3	TDO	From target system	30pF load
4	GND	- n/a -	
5	TCK	From USB TAP probe connector	50mA driver
6	GND	- n/a -	

Table continues on the next page...

Table 8-1. USB TAP probe signal directions (continued)

OnCE pin	Signal mnemonic	Signal direction	Description
7	Reserved	From USB TAP probe connector	50mA driver
8	No-Connect/Key	- n/a -	
9	$\overline{\text{RESET}}$	Bi-directional	Open-drain. 100Ohm to ground when asserted by USB TAP probe, 35pF load when not asserted
10	TMS	From USB TAP probe connector	50mA driver
11	VDD	From target system	2MOhm pull-down, plus 0.01uF load
12	Reserved	Bi-directional	Open-drain, 100Ohm to ground when asserted by USB TAP probe, 35pF load when not asserted[1]
13	Reserved	Bi-directional	Open-drain, 100Ohm to ground when asserted by USB TAP probe, 35pF load when not asserted[1]
14	$\overline{\text{TRST}}$	From USB TAP probe connector	50mA driver

1. 4.7KOhm pull-up to buffered VDD.

NOTE

All OnCE signals must meet accepted standards for OnCE signal design. To ensure proper and stable operation between the USB TAP probe and the target, the OnCE signals must meet the requirements listed in table below.

The following table provides a general description of each OnCE signal and the operational requirements.

Table 8-2. USB TAP probe for OnCE signal recommendations and requirements

OnCE pin	Signal mnemonic	Requirement
1	TDI	Must be wired to the target processor. The USB TAP probe drives the TDI output with up to 50 mA. The TDI trace should be kept short and maintain a "two-signal-width" spacing from any other parallel dynamic signal trace. TDI should have an RC termination option at the processor.
2	GND	Must be wired to the target. GND is connected directly to the USB TAP ground inside the USB TAP probe.

Table continues on the next page...

Table 8-2. USB TAP probe for OnCE signal recommendations and requirements (continued)

OnCE pin	Signal mnemonic	Requirement
3	TDO	Must be wired to the target processor. TDO is an output from the target processor and input to the USB TAP probe. The TDO trace run should be kept short and maintain a "two-signal-width" spacing from any other parallel dynamic signal trace. TDO should have a series termination resistor located near the target processor.
4	GND	Must be wired to the target. GND is connected directly to the ground inside the USB TAP probe.
5	TCK	Must be wired to the target processor. The USB TAP probe drives the TCK output with up to 50 mA. The TCK trace run should be kept as short as possible and maintain a "two-signal-width" spacing from any other parallel dynamic signal trace.
6	GND	Must be wired to the target. GND is connected directly to the ground inside the USB TAP probe.
7	Reserved	Not required for emulation.
8	No-Connect/Key	Not required for emulation. Pin 8 should be clipped on the target OnCE header.
9	$\overline{\text{RESET}}$	Must be wired to the target processor. The USB TAP probe drives the $\overline{\text{RESET}}$ output with 50mA. During reset, the USB TAP probe drives $\overline{\text{RESET}}$ to ground through a 100Ohm resistor.
10	TMS	Must be wired to the target processor. The USB TAP probe drives the TCK output with up to 50mA. The TCK trace run should be kept as short as possible and maintain a "two-signal-width" spacing from any other parallel dynamic signal trace.
11	VDD	Must be wired to the target. The USB TAP probe uses this signal to determine if power is applied to the target. This signal is also used as a voltage reference for the signals driven by the USB TAP probe (TDI, TCK, TMS, $\overline{\text{RESET}}$, and $\overline{\text{TRST}}$).
12	Reserved	Not required for emulation.
13	Reserved	Not required for emulation.
14	$\overline{\text{TRST}}$	Must be wired to the target processor. The USB TAP probe drives the $\overline{\text{TRST}}$ output with up to 50 mA. The $\overline{\text{TRST}}$ trace run should be kept short and

Table 8-2. USB TAP probe for OnCE signal recommendations and requirements

OnCE pin	Signal mnemonic	Requirement
		maintain a "two-signal-width" spacing from any other parallel dynamic signal trace.

Chapter 9
USB TAP probe firmware (Loader)

This chapter explains the methods for reprogramming the loader image stored in the flash EPROM of the USB TAP probe.

This chapter contains the following sections:

9.1 USB TAP probe internal software overview

This section provides an overview of the USB TAP probe internal software.

9.1.1 Loader software

The loader software provides initial USB connectivity to the host, tools for reprogramming the USB TAP loader firmware, and the underlying software framework required to run the debugger.

When the USB TAP probe is first connected to the host USB port it runs the loader software. This is indicated by the red or green blinking heartbeat of the probe's TX/RX indicator light. To reprogram the loader image stored in the probe's flash EPROM, see Reprogramming USB TAP firmware image topic.

9.1.2 Dispatcher software

The USB TAP dispatcher software is basically transparent to the user, and is simply the application that tells the USB TAP probe how to control the target. It recognizes the specific target processor and debug port interface, and carries out the instructions of the debugger.

9.2 Reprogramming USB TAP firmware image

At some point you may be required to reprogram the USB TAP probe's firmware image stored in its flash EPROM. Typically this occurs when you are installing an update to existing software, and the release letter specifies a later version of probe core software.

9.2.1 To reprogram the USB TAP firmware image

1. Make sure the USB TAP probe is connected to a USB Port on your host computer.
2. Launch CCS and open the CCS command window. The procedure is slightly different on Windows and Linux host machines.
   - For Windows, run the command:

     ```
     CodeWarrior_Installation\ccs\bin\ccs.exe
     ```

     This will launch CCS and add a CCS icon (see below figure) to your taskbar. Double-click that icon in the taskbar to open the command window.

   - For Linux, run the command:

     ```
     CodeWarrior_Installation/ccs/bin/ccs
     ```

     This will launch CCS and open the command window automatically.

Figure 9-1. CCS icon

3. In the CCS Command window, enter the commands:

   `delete all updateutap [`*serial_number*]

   The *serial_number* parameter is the serial number of your USB TAP probe, which is shown on the bottom of the device. If you have only one USB TAP probe connected to your host machine, you do not need to specify the *serial_number* parameter. Output similar to the following will be displayed.

```
USB TAP Loader current software ver. {1.0}
Sending code to USB TAP - please wait

*** WARNING ***

DO NOT UNPLUG YOUR USB TAP: Doing so will render your USB

TAP non-functional and require factory reprogramming.

Please wait until the flashing Yellow/Green TGT STATUS light
turns off.

*** WARNING ***
```

4. Before disconnecting your USB TAP probe, wait for the flashing status light to turn off.

9.3 What to do next

When you have completed reprogramming the firmware images stored in the USB TAP flash EPROM, configure your debugging environment (if you have not already done so) as explained in the debugger manual.

Chapter 10
Troubleshooting

If you are having problems with Linux permissions during the installation of the CodeWarrior software on your Linux operating system, please try the following:

- If you are not able to connect to the USB TAP probe, run CCS as root. Run the command:

  ```
  CodeWarrior_Installation/ccs/bin/ccs
  ```

  Leave this instance of CCS running and try connecting using the CodeWarrior software. If this works then there is probably an issue with the USB TAP probe permissions.

- The CodeWarrior installation includes hotplug scripts which should set the appropriate permissions for the USB TAP probe.
  - To confirm that they have been installed correctly, ensure that the file `/etc/hotplug/usb/usbtap` exists and that it is executable by root.
  - View the file `/etc/hotplug/usb.usermap` and ensure that it contains a `usbtap` entry.
  - On Mandrake®Linux® 10.0, ensure that the file `/usr/lib/hotplug/usbtap/usb.usermap` exists, and is readable by root but not executable by root.
  - If the scripts are not installed correctly, reinstall the CodeWarrior software. Otherwise run the following command and send the output to Customer Support with a description of the problem:

    ```
    grep usb.agent /var/log/messages
    ```

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number UTAPUG
Revision 10.x, 06/2014