CodeWarrior TAP Probe User Guide

Revised: 3 October 2012

freescale™

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

How to Contact Freescale

Corporate Headquarters	Freescale Semiconductor, Inc. 6501 William Cannon Drive West Austin, Texas 78735 U.S.A.
World Wide Web	`http://www.freescale.com/codewarrior`
Technical Support	`http://www.freescale.com/support`

Table of Contents

1 Introducing the CodeWarrior TAP Probe 5

1.1 What is CodeWarrior TAP Probe? 5
1.1.1 Product Highlights 6
1.1.2 The Debugging Environment 7
1.1.3 CodeWarrior TAP Probe Benefits 7
1.1.4 Target Connections 7
1.2 Operating Requirements 8
1.2.1 Standard Electrostatic Precautions 8
1.2.2 Operating Temperature 8
1.2.3 Electrical Requirements 8
1.3 Related Documentation 10

2 Connecting to a Network 11

2.1 Connecting the CodeWarrior TAP Probe to the Network 11
2.2 Customizing the CodeWarrior TAP Probe 13
2.3 Testing Network Communication 15

3 Connecting to the Target System 17

3.1 Debug Port Connector Information 17
3.2 Connecting to the Target System 18
3.2.1 Connecting Probe Tip to the Target 18
3.3 Connecting to the Target System Serial Port 19
3.3.1 Connecting the CodeWarrior TAP Probe to the Target System 20
3.3.2 Configuring the Target Serial Port 20
3.3.3 Accessing the Target Serial Port 22

4 Using the CodeWarrior TAP Probe 23

4.1 Debugging with the CodeWarrior TAP System 23
4.1.1 Run/Pause/Mixed Mode States 24
4.1.2 Connecting to Multiple TAP Probes 24
4.2 Accessing a Ethernet TAP Probe Remotely 25

5 Hardware Specifications 27
5.1 LEDs on CodeWarrior TAP Probe 27
5.1.1 Transmit/Receive Indicator. 28
5.1.2 Run/Pause Indicator 28
5.1.3 RJ45 Ethernet Connector with Link and Activity Indicators 29
5.2 Host Connectors on CodeWarrior TAP Probe. 29
5.2.1 RJ45 Ethernet Connector 30
5.2.2 USB Connector. 30
5.3 Target Connectors on CodeWarrior TAP Probe 30
5.3.1 RJ25 Target Serial Connector. 30
5.3.2 Probe Tip Connector. 31
5.3.3 Electrical Characteristics 32
5.3.4 Physical Considerations 32

A Ethernet TAP Probe Setup Utility Commands 35
A.1 Connecting to the CodeWarrior TAP Probe Setup Utility 35
A.2 Ethernet TAP Probe Setup Utility Commands and Variables 36
A.2.1 Commands to Configure Communications .36
A.2.2 Commands to Troubleshoot Communication 40

B Network Administration 43
B.1 CodeWarrior TAP Probe Network Ports. 43
B.2 Configuring the Ethernet TAP Probe Using netparam 44
B.2.1 Configuring a Dynamic IP Address. 44
B.2.2 Configuring a Static IP Address. 44
B.2.3 Static Routing 45
B.2.4 Changing an Existing Route Entry 46
B.2.5 Entering Static Routes 47
B.3 Using CCS to Search for CodeWarrior TAP Probes. 48

C Ethernet TAP Probe Firmware (Core) 51
C.1 CodeWarrior TAP Probe Internal Software Overview 51
C.1.1 Boot Loader 51
C.1.2 Operating System. 51

C.1.3 Shell Software 52
C.2 Reprogramming the Ethernet TAP Probe Firmware Images 52
C.2.1 Reprogramming the Firmware through the Ethernet Port 52

D JTAG/COP Connector Information 55

E OnCE Connector Information 61

F ColdFire BDM Connector Information 65

G Troubleshooting 71

G.1 Troubleshooting Communications Problems 71
G.1.1 Verify Network Communication 71
G.1.2 View Network Connections 72
G.2 Troubleshooting Power Problems 72

Index 75

1

Introducing the CodeWarrior TAP Probe

The CodeWarrior TAP probe allows your personal computer workstation to communicate with Freescale Power Architecture, StarCore, and ColdFire processors using a privileged debug connection, such as COP, OnCE, or BDM.

This chapter explains:

- What is CodeWarrior TAP Probe?
- Operating Requirements
- Related Documentation

CAUTION The CodeWarrior TAP probe contains components that are subject to damage from electrostatic discharge. Whenever you are using, handling, or transporting the CodeWarrior 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.

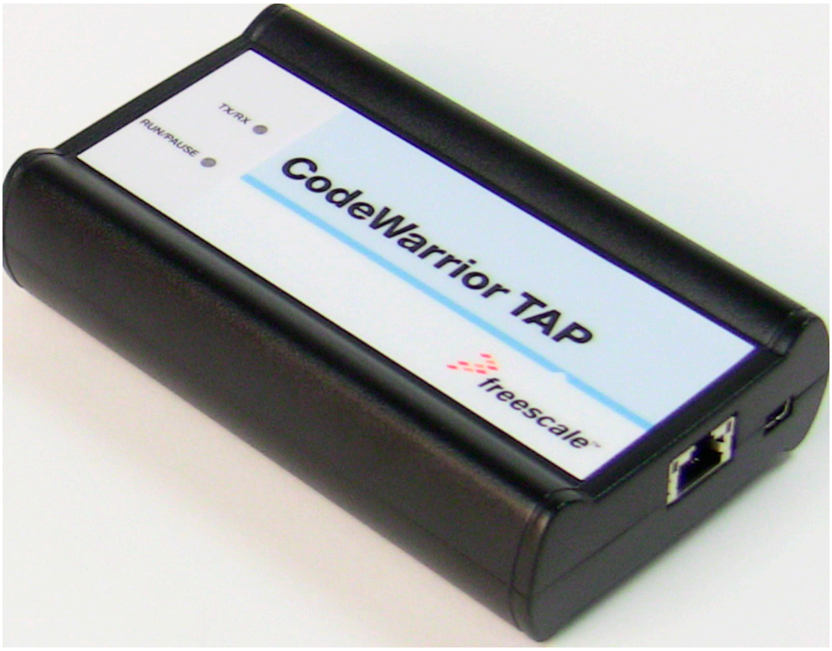
1.1 What is CodeWarrior TAP Probe?

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

The basic CodeWarrior TAP probe system is composed of two parts:

- The CodeWarrior TAP probe (Figure 1.1), which provides visibility into and control of your target system using a JTAG or BDM interface, and connects to your host computer through a USB 2.0, 10BaseT, or 100BaseT link.
- JTAG/COP, StarCore, or ColdFire/BDM probe tip, which is designed to provide a physical and electrical interface to the target system processor that you want to gain visibility into.

Figure 1.1 CodeWarrior TAP Probe



1.1.1 Product Highlights

The CodeWarrior TAP probe has these features:

- Supports the Power Architecture, StarCore, and ColdFire™ processors. Go to the http://freescale.com/CWTAP for the latest supported Freescale processors
- Supports all CPU core speeds
- Allows you to control and debug software running in-target, with minimal intrusion into the target system operation
- Allows you to debug code in cache, ROM, RAM, and flash memory
- Supports 10/100 Ethernet network connection
- Supports telnet access to your target systems serial port, allowing you to interact with your target system’s serial port over the network
- Supports USB 2.0 high-speed connection
- USB powered
- Supports both big and little endian byte-order

- Software debug capabilities, usually part of host software like CodeWarrior, include:
  - Control instruction execution
  - Display and modify target system memory
  - Examine and modify any processor registers
  - Run to breakpoints in ROM, RAM, or flash memory
  - Single-step through source and assembly language code views
  - Step into, over, or out of functions
  - Collect and analyze real-time data
  - Perform boundary scan testing with support from correct host-level software
  - Program all onboard memories with support from correct host-level software

1.1.2 The Debugging Environment

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

1.1.3 CodeWarrior TAP Probe Benefits

The CodeWarrior TAP probe provides these key benefits:

- *Visibility*: Allows you to observe registers and the current state of target system memory. You can halt program execution at predefined states and examine the data for a particular program state.
- *Control*: Enables you to 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 TAP probe connects to your target through the standard debug port for the processor family, and supports a single target connection, based on the connected probe tip. For details on processor list, go to http://freescale.com/CWTAP.

The TAP probes are available in the following Freescale versions:

- JTAG/COP for Power Architecture™, QorIQ, PQII, PQIII, AMP, Qonverge, but not PQI
- BDM for ColdFire® targets (not ColdFire v1)
- OnCE for StarCore

For information on connecting to a target, refer to the Connecting to the Target System chapter.

1.2 Operating Requirements

Before setting up the system, ensure 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.

It is recommended that you use the following precautions:

- Use wrist straps or heel bands with a 1 MΩ resistor connected to ground.
- On the work surface and floor, use static conductive mats with a 1 MΩ 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 Operating Temperature

For operating temperature of TAP probe, refer to the Physical Considerations topic.

1.2.3 Electrical Requirements

The TAP probe can be powered through a USB cable and does not require an external power supply. It is designed to be plugged directly into a host computer, but can also work with self-powered hubs. For details on Bus-powered hubs, refer to the Electrical Characteristics topic. If your hub is not able to provide sufficient power, connect the TAP probe directly to your host PC, or purchase a self-powered USB hub.

If you only plan to use Ethernet communications, the CodeWarrior TAP probe can be powered from the external power supply provided with your unit. It can use line voltages of 100-240 VAC (50/60 Hz).

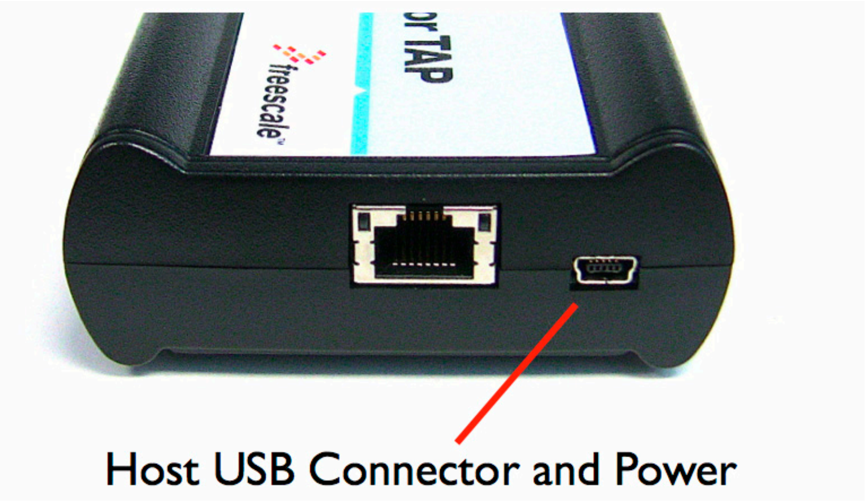
NOTE It is recommended to use a surge protector between the power supply and AC power.

1.2.3.1 Connecting the Power Supply Cable

Connect the power supply USB cable connector to the USB connector on the CodeWarrior TAP probe as shown below.

CAUTION Connect only the provided power supply to the CodeWarrior TAP probe. Other power supplies may look similar, but can damage the probe if the supply specifications differ from the required specifications.

Figure 1.2 CodeWarrior TAP Probe with USB Connector



1.2.3.2 Cycling Power to the System

When you need to apply or cycle power to the CodeWarrior TAP probe, connect or disconnect the power cable from the power source or from the probe. After you have connected the probe to your target system, use the following sequence for applying or removing the power.

To turn the power on

1. Turn on the CodeWarrior TAP probe power.
2. Turn on the target system power.

To turn the power off

1. Turn off the target system power.
2. Turn off the CodeWarrior TAP probe power.

NOTE In the case of PowerPC targets with a $\overline{\text{QACK}}$ signal, for the CodeWarrior TAP probe to properly stop and restart the target, the $\overline{\text{QACK}}$ signal must be pulled low. The CodeWarrior TAP probe pulls this signal low through the probe tip.

1.3 Related Documentation

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

2

Connecting to a Network

This chapter explains how to connect the CodeWarrior TAP probe to an existing TCP/IP network.

The CodeWarrior TAP probe is an Ethernet device that may be configured for TCP/IP either using DHCP to acquire its IP configuration (the default method) or through a static IP configuration.

This chapter explains:

- Connecting the CodeWarrior TAP Probe to the Network
- Customizing the CodeWarrior TAP Probe
- Testing Network Communication

2.1 Connecting the CodeWarrior TAP Probe to the Network

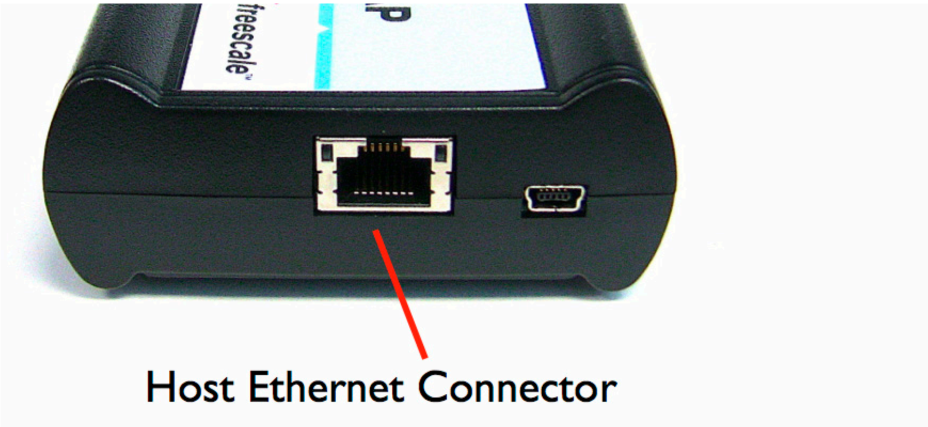
The CodeWarrior TAP probe's default operation is to acquire its network configuration automatically using DHCP, and attempt to register its hostname with a name server. The factory assigned host name is FSL*XXYYZZ* where *XXYYZZ* is the last three octets of the Ethernet MAC address, provided on a label on the bottom side of the probe. For example, if the probe's Ethernet MAC address is 00:04:9f:00:77:31, the host name will be FSL007731. Figure 4.1 shows TAP serial number.

The CodeWarrior TAP probe can connect directly to a network using Ethernet (10/100BaseT) cables.

To connect to the Ethernet interface

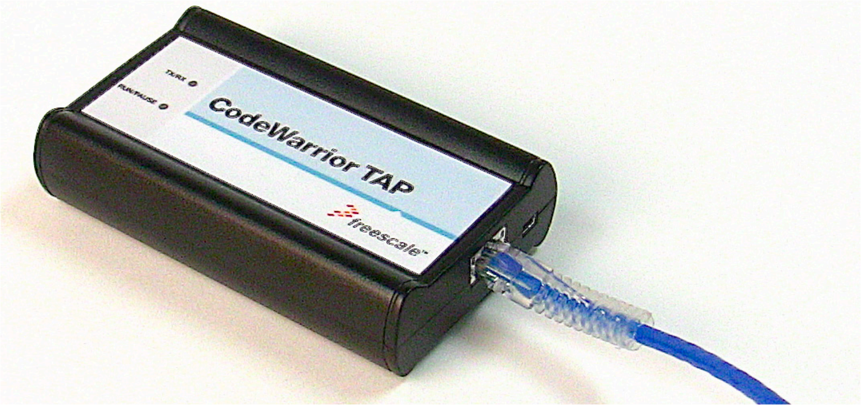
1. Plug one end of the supplied RJ45 cable (p/n 600-75499) into the RJ45 connector of the CodeWarrior TAP probe.

Figure 2.1 CodeWarrior TAP Probe with Ethernet Connector



2. Connect the other end of the RJ45 cable into the RJ45 connector of the Ethernet network or host computer.

Figure 2.2 CodeWarrior TAP Probe with an RJ45 Cable Attached



NOTE When you configure the debugger for the hardware connection, you will need to specify the CodeWarrior TAP probe IP address or hostname. The CCS `findcc` utility is used to search any probe on the local subnet. For more information, refer to the Using CCS to Search for CodeWarrior TAP Probes topic.

NOTE Depending on the type and complexity of your network, your network administrator may need to update network server tables so that the network accesses the CodeWarrior TAP probe correctly. Updating network server tables requires both a detailed knowledge of Ethernet address resolution and network routing with write access permission to the server tables. For more information on network administration, refer to the Network Administration topic.

2.2 Customizing the CodeWarrior TAP Probe

As shipped, the CodeWarrior TAP probe acquires its network configuration automatically using DHCP. If you cannot use DHCP, you must configure the probe for your network using static IP address resolution.

To manually configure the network settings of the CodeWarrior TAP probe for your network, access the probe configuration console as described below and use the probe on-board setup utility `netparam` to change the probe network settings. The probe `netparam` utility lets you select and modify network parameters that are saved in probe memory. Use `netparam` to configure the probe to match the network address resolution and routing protocols.

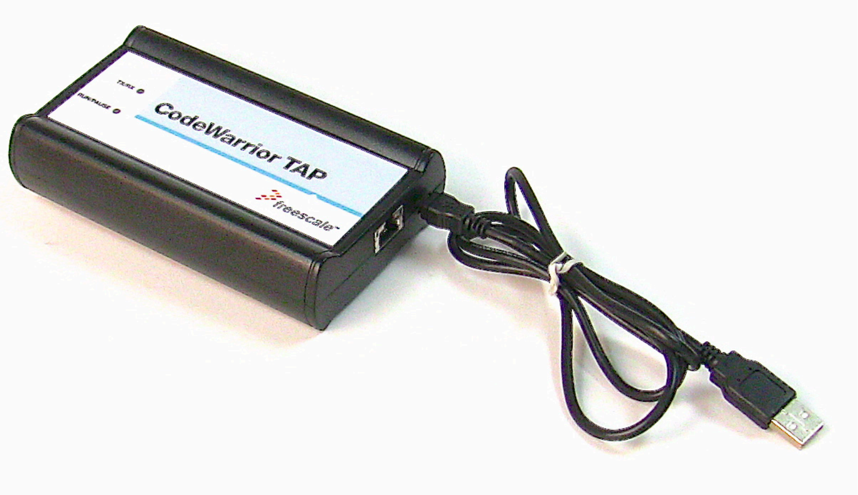
If the probe needs to communicate with hosts on other subnets, you will need to configure the probe for one of the following routing options:

- Default gateways
- Static routing tables

To Access the CodeWarrior TAP Configuration Console

1. Connect the other end of the USB cable to the USB connector of the CodeWarrior TAP probe (Figure 2.3).

Figure 2.3 CodeWarrior TAP Probe with USB Cable Attached



2. Wait for the TX/RX LED to start flashing green.
3. Identify the serial port device assigned to the CodeWarrior TAP. On Windows, click **Start > Control Panel >Adminstrative Tools > Computer Management > Device Manager > Ports** and then select **USB serial port** from the ports list. On Linux, the device file is located at: `/dev/ttyACM0`.
4. When prompted, press **Enter**. The login banner should be displayed and the `core>` command-line prompt appears.

To customize the CodeWarrior TAP probe network settings

1. Change the CodeWarrior TAP probe network settings.
   a. At the `core>` prompt, enter the `netparam` command to view the current settings.
   b. For network setup, see the netparam section for syntax and options. For more information on installing the CodeWarrior TAP probe on a network, see the Network Administration section.
   c. At the `core>` prompt, enter the `netparam` commands and required parameters.

2. At the core> prompt, enter reset to reboot the CodeWarrior TAP probe to activate the new network settings.

NOTE If you connect to the CodeWarrior TAP using telnet rather than the USB configuration console, you may lose access when you change network settings, and will need to reconnect after the settings have changed.

Example: Assign a static IP address and hostname to the CodeWarrior TAP probe

If the CodeWarrior TAP probe has a static IP address of 195.121.1.2 and a hostname of lab01, enter the following commands:

```
core> netparam static_ip_address 195.121.1.2

core> netparam bootconfig static:lab01

core> reset
```

The netparam utility copies its settings into non-volatile memory on the probe. Follow these rules while using the netparam utility:

- Each time you enter a netparam command, wait for the core> prompt to re-appear before entering the next command. The prompt indicates that the parameter change is logged.
- When you have finished entering all settings, type reset at the core> prompt. When the probe restarts, it will use the new netparam parameters.

2.3 Testing Network Communication

You can use the ping command to ensure that the CodeWarrior TAP probe can communicate with the host.

To verify communication

At a host command prompt, type the following:

```
ping hostname | ip_address
```

where *hostname* is the name and *ip_address* is the IP address assigned to the CodeWarrior TAP probe.

If no output is displayed on the screen, check the following:

- The physical connections are tight.
- The CodeWarrior TAP probe address and netmask in the hosts file match those in CodeWarrior TAP probe flash.
- The netmask used for the CodeWarrior TAP probe and for the Ethernet Network Interface Card (NIC) are appropriate to the class of the IP address.

3

Connecting to the Target System

To use your CodeWarrior TAP probe, you must have a prototype hardware or an evaluation board.

This chapter explains how to connect a CodeWarrior TAP probe to the target system.

This chapter explains:

- Debug Port Connector Information
- Connecting to the Target System
- Connecting to the Target System Serial Port

3.1 Debug Port Connector Information

The CodeWarrior TAP probe offers debugging capabilities without modifying any target system code or any special I/O port in the target system for communication with a monitor program running on the target system. Target system connections can be made using the debug ports (JTAG/COP, StarCore, or ColdFire BDM).

The CodeWarrior TAP probe connects to the target system's JTAG header using a probe tip adapter and ribbon cable. The TAP probe is a powerful development tool for use with a wide variety of processors that use either JTAG/COP, ColdFire BDM, or StarCore debug interfaces.

The following appendices describe the debug port connector specifications:

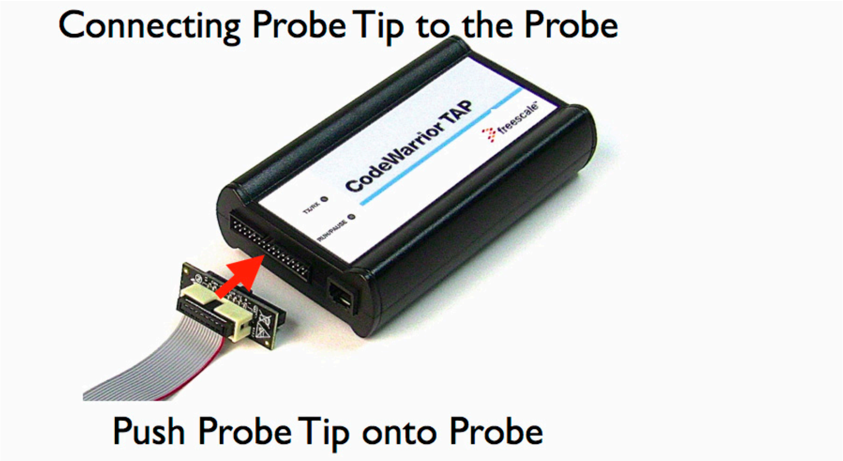
- "JTAG/COP Connector Information"
- "ColdFire BDM Connector Information"

3.2 Connecting to the Target System

CAUTION Failure to properly connect the 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 connect to the CW TAP probe. Make sure that you properly align the TAP multi-pin socket connector with the multi-pin header on your target system.

Figure 3.1 CW TAP Probe — Connecting Probe Tip to the Probe



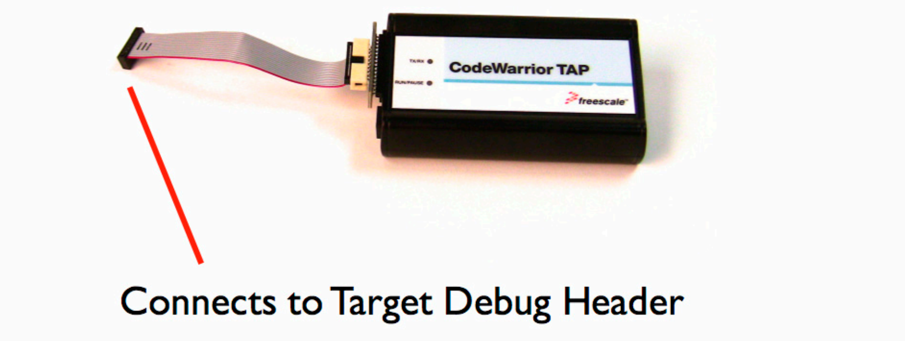
NOTE Pin 1 is clearly marked on the gray ribbon cable by a red line down one side of the cable and a small triangle in the plastic socket.

3.2.1 Connecting Probe Tip to the Target

To connect the CW TAP cable to the target debug port header:

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

Figure 3.2 CW TAP Probe — Connecting to the Target

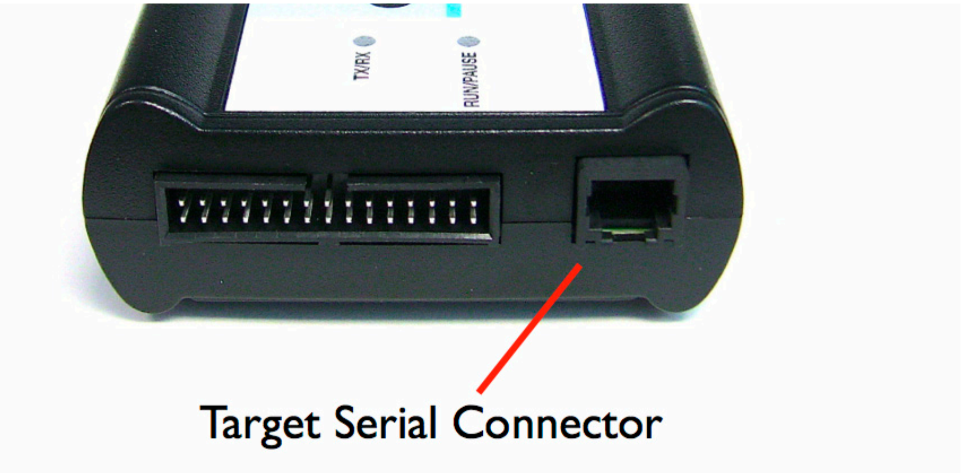


5. Gently (but firmly) press the connector onto the target system debug port header.

3.3 Connecting to the Target System Serial Port

Many target system boards have a built-in serial port. A console interface connection to the serial port of the target system lets you query and configure the state of your target system.

Figure 3.3 CW TAP Probe — Target Serial Connector



The CodeWarrior TAP probe provides a serial port which can be configured to access the serial port of the target system. This is useful if you need to access the serial port of a remotely located target system over Ethernet from the host system.

The following sections explain how to access the serial port of the target system:

- Connecting the CodeWarrior TAP Probe to the Target System
- Configuring the Target Serial Port
- Accessing the Target Serial Port

3.3.1 Connecting the CodeWarrior TAP Probe to the Target System

A RJ25 cable (P/N 600-76822) is provided with the CodeWarrior TAP probe to connect to the serial port of your target system.

To connect the serial cable between the CodeWarrior TAP probe and the serial port of the target system

1. Connect one end of the RJ25 cable and the appropriate adapter to the serial port on your target system board.
2. Connect the other end of the RJ25 cable to the CodeWarrior TAP probe RJ25 serial connector.

3.3.2 Configuring the Target Serial Port

This table shows the default settings of the CodeWarrior TAP Target Serial port.

Table 3.1 CodeWarrior TAP Probe Target Serial Port Default Settings

For this option...	Select...
Baud rate	9600
Data bits	data8
Stop bits	stop1
Parity	noparity
Hardware flow control	nortscts
XON/XOFF flow control	noxon
Target echo feature	echo

If the CodeWarrior TAP probe Target Serial port settings do not match the serial port settings of your target system, use the following steps:

To configure the CodeWarrior TAP probe serial port

1. Make sure network communications are configured correctly. For more information, refer to the Connecting to a Network chapter or Setting Up a Standalone PC Ethernet appendix.
2. Connect to the CodeWarrior TAP probe internal setup utility. For more information, refer to the Connecting to the CodeWarrior TAP Probe Setup Utility topic.
3. When the `core>` prompt appears on the terminal, enter the `tgtty` command to configure the CodeWarrior TAP probe Target Serial port. The syntax is:

   ```
   tgtty
     [9600|19200|38400|57600|115200]
     [data8|data5|data6|data7]
     [stop1|stop2]
     [noparity|oddparity|evenparity|lowparity|highparity]
     [rtscts|nortscts]
     [xon|noxon]
     [echo|noecho]
   ```

   For example:

   ```
   tgtty 19200 data8 stop2 noparity nortscts noxon echo
   ```

4. Verify the Target Serial port configuration at the `core>` prompt by entering the `tgtty` command by itself:

   ```
   tgtty
   ```

To restore the target serial port to the default settings

1. Make sure network communications are configured correctly. For more information, refer to the Connecting to a Network chapter or Setting Up a Standalone PC Ethernet appendix.
2. Connect to the DCU probe internal setup utility. For more information, refer to the Connecting to the CodeWarrior TAP Probe Setup Utility topic.
3. When the `core>` prompt appears on the terminal, use the `tgtty` command to reset the Target Serial port to the default settings:

   ```
   tgtty default
   ```

3.3.3 Accessing the Target Serial Port

You can use telnet to connect to the CodeWarrior TAP probe Target Serial port and access the serial port of your target system remotely over Ethernet.

To telnet to the Target Serial port

1. Make sure that you have physically connected the DCU probe RJ25 cable to your target system (for more information, refer to the Connecting the CodeWarrior TAP Probe to the Target System topic).
2. Verify the serial port settings (refer to the Configuring the Target Serial Port topic).
3. Start a telnet session and connect to the DCU probe Target Serial port:

   ```
   telnet {hostname | ip_address} 1082
   ```

   Use the host name or IP address of the probe. For static IP, the host name must be the same one you entered into the `hosts` database file; refer to the Connecting to a Network chapter or Setting Up a Standalone PC Ethernet appendix. To identify the IP address of any probe on the subnet, refer to the Using CCS to Search for CodeWarrior TAP Probes topic. The Target Serial port number of the CodeWarrior TAP probe is 1082.
4. You should now have access to the serial port of your target system. You can use this connection in the same manner as if your host computer were connected directly to the serial port of your target system.

4

Using the CodeWarrior TAP Probe

This chapter provides system startup procedures and explains how the CodeWarrior TAP probe is accessed remotely.

This chapter contains the following sections:

- Debugging with the CodeWarrior TAP System
- Accessing a Ethernet TAP Probe Remotely

4.1 Debugging with the CodeWarrior TAP System

This section explains how to start debugging with the CodeWarrior TAP probe.

Before starting debug with the CodeWarrior TAP probe, make sure you have:

- Connected the CodeWarrior TAP probe to your network or computer.
- Connected the CodeWarrior TAP probe to the target system.
- Installed the debugger software such as CodeWarrior Development Studio and properly configured it to communicate with the CodeWarrior TAP probe.

To start the CodeWarrior TAP probe

1. Apply power to the CodeWarrior TAP probe.
2. Apply power to the target system.
3. Start the CW debugger.
4. Configure the debugger for the CodeWarrior TAP connection.

LEDs are provided to indicate the status of the CodeWarrior TAP probe. For details on the LED indicators, refer to the CodeWarrior TAP Probe Specifications topic.

You are now ready to begin your debug session. For information on using the debugger, refer to the *Targeting User Guide*.

The following topics provide information specific to TAP probe operation:

- Run/Pause/Mixed Mode States

- Connecting to Multiple TAP Probes

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

4.1.1 Run/Pause/Mixed Mode States

When the host debugger is connected to the target using the 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 — The Run/Pause LED will be green. In this mode, all target system processor cores execute the target code.
- Pause mode — The Run/Pause LED will be red. In this mode, all target system processor cores have stopped executing the target code.
- Mixed mode — The Run/Pause LED will be orange. In this mode, some target system processor cores are in run mode and others are in pause mode.

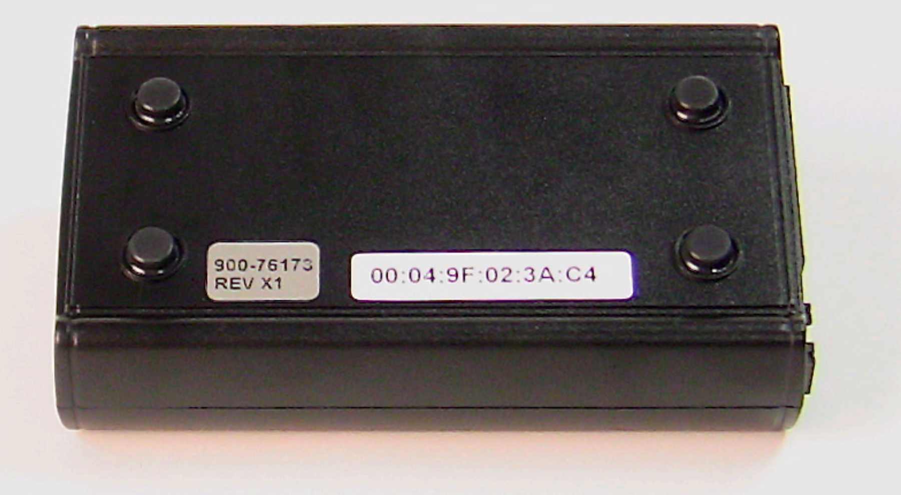
4.1.2 Connecting to Multiple TAP Probes

You can connect to multiple 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 TAP probe connections in the IDE, simply define the connections, entering the unique probe serial number for each device. The IDE will manage the CodeWarrior Connection Server (CCS) sessions. The host that has CodeWarrior installed issues read and write action to the memory through the connection protocol called CCS.
- For tools that do not support creating multiple 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 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 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.

Figure 4.1 CodeWarrior TAP Probe — Bottom view



TIP To set up the debug connection, you will need to know the 12-digit TAP probe serial number, located on a label on the bottom of the device.

4.2 Accessing a Ethernet TAP Probe Remotely

You can remotely access the internal setup utility and the Target Serial port of the DCU probe after you connect the probe to your network.

If the host computer is not physically located near the DCU probe, remote access is useful when you need to:

- Reconfigure communications
- Use the serial port of your target system
- Reset the DCU probe through your Ethernet connection

To remotely access the setup utility

Open a telnet session and connect to the DCU probe by entering the command:

```
telnet hostname | ip_address
```

Use the host name or IP address of the probe. For static IP, the host name must be the same one you entered into the `hosts` database file; refer to the Connecting to a Network chapter or Setting Up a

Standalone PC Ethernet appendix. To identify the IP address of any probe on the subnet, refer to the Using CCS to Search for CodeWarrior TAP Probes topic.

The login banner is displayed, followed by the `core>` command-line prompt.

To connect to your target's serial port remotely

Make sure the DCU probe Target Serial port is physically connected to your target's serial port, and it is configured correctly. For more information, refer to the Accessing the Target Serial Port topic.

5

Hardware Specifications

This chapter provides hardware specifications for the DCU probe.

This chapter contains the following sections:

- LEDs on CodeWarrior TAP Probe
- 5.2Host Connectors on CodeWarrior TAP Probe
- Target Connectors on CodeWarrior TAP Probe
- CodeWarrior TAP Probe Specifications

5.1 LEDs on CodeWarrior TAP Probe

This figure shows the various LEDs of the CodeWarrior TAP probe.

Figure 5.1 CW TAP Probe — LED Indicators

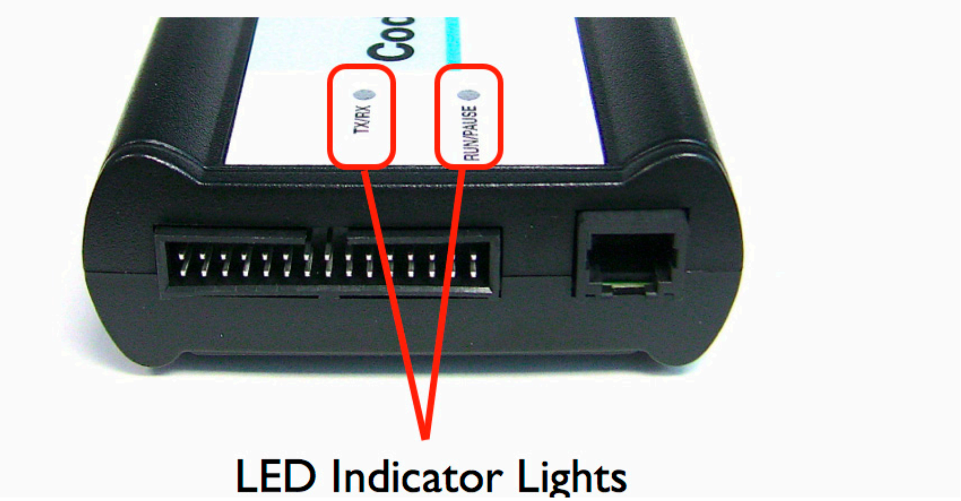
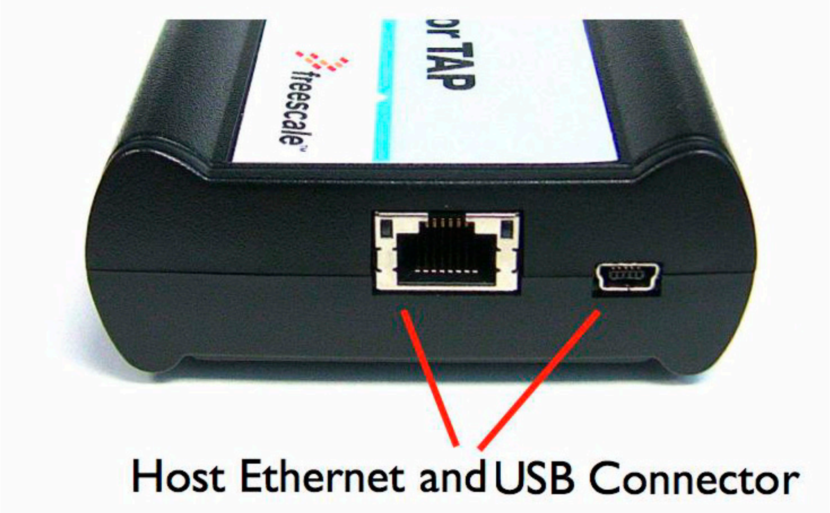


Figure 5.2 CW TAP Probe — Ethernet and USB Connector



5.1.1 Transmit/Receive Indicator

The Transmit/Receive LED (labeled TX/RX) indicates the status of communication between the DCU probe and the network/host as follows:

- The LED is red until the DCU probe boot code starts running.
- The LED flashes orange (1 Hz) during configuration of the network/USB interface.
- The LED flashes green (1 Hz) after network/USB interface has is successfully configured. During firmware updates, the LED flashes green at a higher frequency (5Hz).

NOTE Do not remove power, unplug the network, or press the reset button during firmware updates.

- The LED flashes orange when the CodeWarrior TAP is communicating with the target.
- The LED is unlit if the DCU probe is not powered on.

5.1.2 Run/Pause Indicator

The status LED (labeled RUN/PAUSE) indicates the state of the target as follows:

- The LED is off when no target power is detected.
- The LED is green when the target is in run mode.
- The LED is red when the target is in pause mode.

- The LED is orange when the target is in mixed mode.
- The LED is initially unlit and remains so until the TX/RX LED starts flashing.

5.1.3 RJ45 Ethernet Connector with Link and Activity Indicators

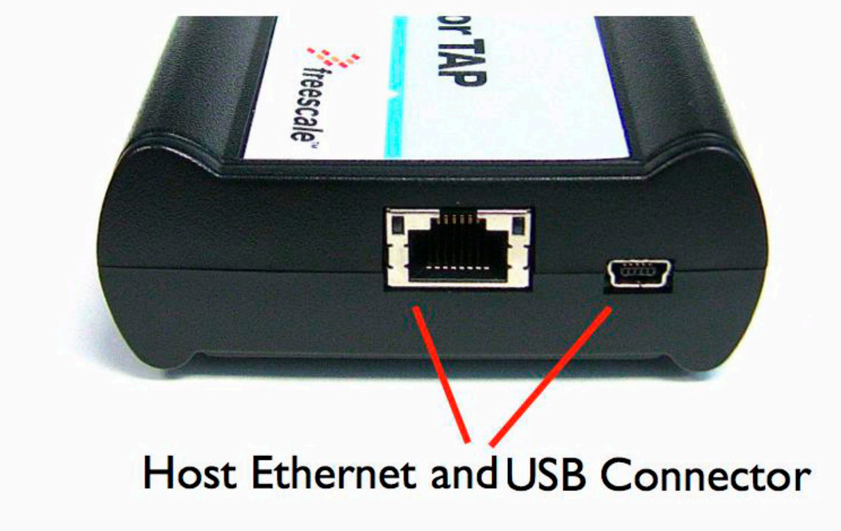
The DCU probe interface consists of an RJ45 connector that connects directly to 10/100 twisted pair networks. See the Connecting to a Network chapter or the Setting Up a Standalone PC Ethernet appendix for more information on connecting to a network.

The CodeWarrior TAP probe link and activity indicators are integrated into the RJ45 CodeWarrior TAP probe connector. The yellow indicator is turned on when the CodeWarrior TAP probe is connected to any network, and flickers when data is being transferred across the network. The green indicator is turned on when the CodeWarrior TAP probe is connected to a 100BaseT network, and flickers when data is being transferred across the network.

5.2 Host Connectors on CodeWarrior TAP Probe

This figure shows the host connectors of the CodeWarrior TAP probe.

Figure 5.3 CW TAP Probe — Host Side View



5.2.1 RJ45 Ethernet Connector

The Ethernet connector on the CW TAP probe is used to connect to a 10/100BaseT Ethernet.

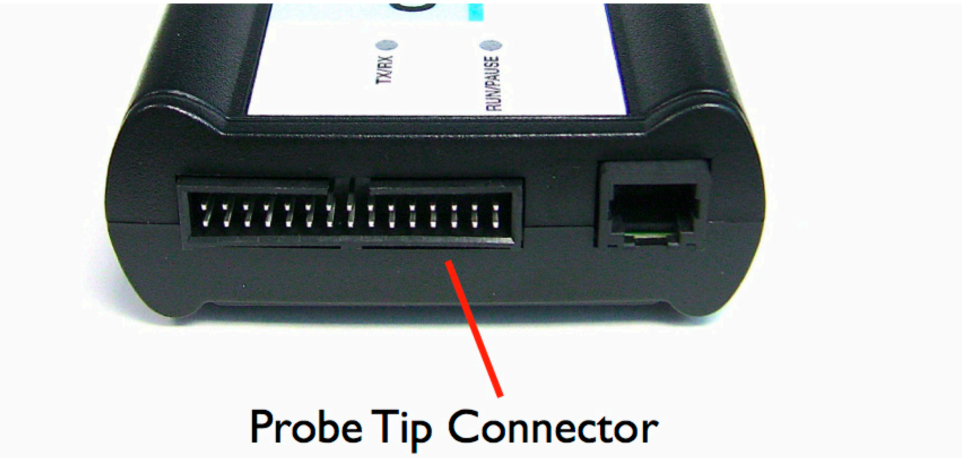
5.2.2 USB Connector

The USB port on the CodeWarrior TAP probe acts as both a virtual serial device and virtual Ethernet device. The virtual serial interface is used for configuring network communication, entering routing tables, and for diagnostics. The virtual Ethernet device is used by the debugger to communicate with the CodeWarrior TAP.

5.3 Target Connectors on CodeWarrior TAP Probe

This figure shows the target connectors of the CodeWarrior TAP probe.

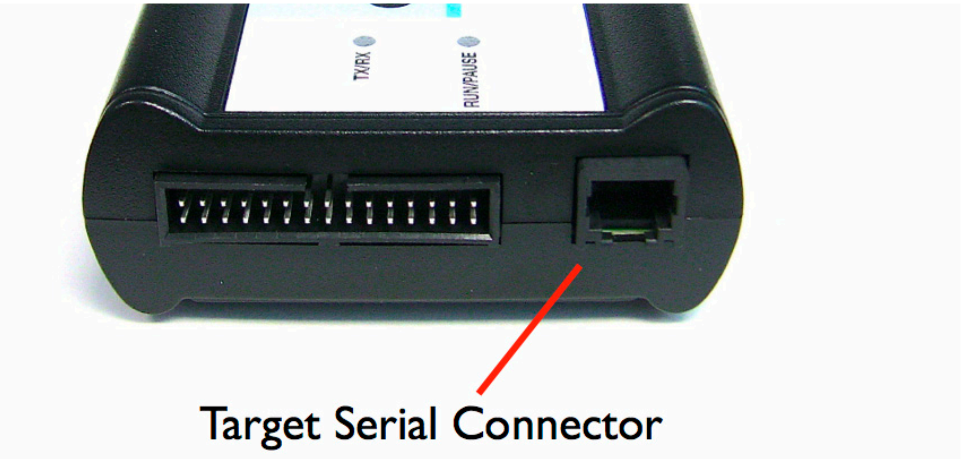
Figure 5.4 CW TAP Probe — Target Side View



5.3.1 RJ25 Target Serial Connector

The DCU probe provides a target serial port which can be configured to access your target’s serial port. This is particularly useful if your host computer is not near your target and you need to access your target’s serial port remotely over your network.

Figure 5.5 CW TAP Probe — Target Serial Connector



This table shows the pinout definition of the Target serial port.

Table 5.1 Pinout Definition of the Target Serial port

Pin	Signal
1	Ready To Send (RTS)
2	Ground
3	Receive Data (RxD)
4	Transmit Data (TxD)
5	Ground
6	Clear To Send (CTS)

Pin 1 is on the right side as you look at the RJ-11 socket (locking tab on the bottom).

5.3.2 Probe Tip Connector

The 30-pin debug port header is used to connect the DCU probe to a debug port header on your target system.

NOTE Ensure that Pin 1 of the probe tip is connected to the Pin 1 of the header.

The probe tip consists of a 6-inch ribbon cable with the appropriate debug adapter attached. The ribbon cable has a red stripe down one side to indicate the location of pin 1.

NOTE The OnCE probe tip cable 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.

5.3.3 Electrical Characteristics

The DCU probe affects the load on only those signals that are connected to the debug port connector. Loading depends on the method used to connect the DCU probe to the target system. Refer to the Connecting to the Target System topic for description of each connection method.

The DCU probe affects the target processor and target electrical characteristics. Caution should be taken in designing the target to accommodate the small signal delays associated with in-circuit emulator or other test equipment.

This table shows the electrical characteristics of the CW TAP probe.

Table 5.2 CW TAP Probe — Electrical Characteristics

Electrical Characteristics	
Target voltage levels supported	1.2V to 3.3V; 5V tolerant
DCU probe power consumption from target	Less than 50 µA to detect target power
USB Power Supply	5V, 500mA

NOTE Bus powered USB hubs are not designed to provide 500mA to devices. The CodeWarrior TAP must be directly connected to a self-powered hub, PC, or the AC adapter included in the kit.

5.3.4 Physical Considerations

This table shows the physical characteristics of the DCU probe.

Table 5.3 Ethernet TAP Probe — Physical Characteristics

Physical Characteristics	
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	
DCU probe dimensions	
Length	5.5” (7.5” with cables)
Width	3.25”
Height	1.375”
Probe tip cable socket dimensions	
Height (above board)	0.375” (0.95 cm)
Thickness	0.20” (0.51 cm)
Pin-to-pin spacing	0.1” (0.25 cm)
Width	Number of Positions x 0.1” (0.25 cm) + 0.18” (0.46 cm)

A

Ethernet TAP Probe Setup Utility Commands

This appendix explains how to access the CodeWarrior TAP probe internal setup utility. It describes all available setup utility commands and arguments.

This appendix contains the following sections:

- Connecting to the CodeWarrior TAP Probe Setup Utility
- Ethernet TAP Probe Setup Utility Commands and Variables

NOTE The commands described in this appendix are for reference only. For detailed procedures on using these commands, refer to the applicable chapter or appendix that covers the topic of interest.

A.1 Connecting to the CodeWarrior TAP Probe Setup Utility

There are two methods for accessing the Ethernet TAP probe internal setup utility:

- Connect to the Ethernet TAP probe USB port. Use this method if the Ethernet TAP probe is not connected to your network. For more information, refer to the Customizing the CodeWarrior TAP Probe topic.
- Telnet to the Ethernet TAP probe through an existing Ethernet connection (refer to the To connect to the setup utility using the telnet port topic).

  Use this method if the Ethernet TAP probe is currently connected to your network.

  Telnet is the Internet standard protocol for remote logins. Most TCP/IP networks provide a telnet program that you can use to login across the network to another machine. Note that if you lose your Ethernet connection by improperly configuring the Ethernet TAP probe from a telnet session, then you will have to connect to the USB port to re-establish network communications.

To connect to the setup utility using the telnet port

1. Open a telnet session and connect to the Ethernet TAP probe.

   ```
   telnet hostname
   ```

   Use the hostname that you entered into the `hosts` database file, as discussed in the Connecting to a Network chapter or the Setting Up a Standalone PC Ethernet appendix.

2. After the login banner is displayed, the `core>` command-line prompt appears.

NOTE Use the `help` command at the `core>` prompt for a list of all the internal Ethernet TAP probe commands available. Or use `help` and the command name for a brief description of the command and a list of the command's arguments.

A.2 Ethernet TAP Probe Setup Utility Commands and Variables

The Ethernet TAP probe internal setup utility commands are for configuration and troubleshooting.

A.2.1 Commands to Configure Communications

The following internal setup utility commands are used to configure the Ethernet TAP probe for network communication (`netparam`), and to communicate with your target system's serial port (`tgtty`).

A.2.1.1 netparam

The `netparam` command displays or sets non-volatile networking parameters stored in the flash EPROM of Ethernet TAP probe. Entered without options, it displays all current settings. To change parameters, specify one or more options. To activate new settings, the unit must be rebooted. For more information on netparam command, refer to the Configuring the Ethernet TAP Probe Using netparam topic.

Syntax

```
netparam
  [add_host host ip_address]
  [add_route host gateway hop_#]
  [bootconfig {static | dhcp }[:host]]
  [delete_host host]
  [delete_route host]
```

```
[static_ip_address address[:mask]]
[static_dns_server address]
```

<null>	
Reports the current configuration	
add_host *host ip_address*	
Adds a hostname-address pair to the static host table. Table entries are automatically entered into the system on reset.	
host	Name to associate with the address *ip_address*
ip_address	IP address to use for *host*, specified in dotted-decimal notation
add_route *host gateway hop_#*	
Adds a route to Ethernet TAP probe static route table. Table entries are automatically entered into the system on reset. If the specified parameters are invalid for the operating network, they are not stored.	
host	Destination IP address of host or host network, specified in dotted-decimal notation. Default is a valid entry for *host*, and equivalent to 0.0.0.0.
gateway	Gateway IP address for probe, specified in dotted-decimal notation
hop_#	Decimal number of gateway hops between Ethernet TAP probe and destination host or network
bootconfig {static | dhcp }[:*host*]	

Sets the IP address resolution protocol. It determines the boot method of Ethernet TAP probe. Use `bootconfig` to connect to the network either by DHCP or by storing the IP address in the flash EPROM of Ethernet TAP probe. When using DHCP, you can specify the host name that you would like the probe to try to register with a name server when it acquires its network configuration. The factory assigned host name is FSL*XXYYZZ*, where *XXYYZZ* is the last three octets of the Ethernet MAC address provided on a label on the bottom side of the probe. For example, if the probe's Ethernet MAC address is 00:00:f6:00:77:31, the default host name will be FSL007731.	
`static`	Use IP address stored in Ethernet TAP probe
`dhcp`	Use the network DHCP protocol to resolve IP address, netmask, and default gateway (default)
`:host`	Host name for the CodeWarrior TAP probe. If `dhcp` is specified, the probe will attempt to register this host name with the DHCP server. There should be no white space before :*`host`*. The ccs `findcc` search utility will report the host name of the probe for both the dhcp and static options.
`delete_host` ***`host`***	
Deletes a hostname-address pair from the static host table	
`host`	Name of host to remove
`delete_route` ***`host`***	
Deletes a route from the static route table	
`host`	Destination IP address of host or host network
`static_ip_address` ***`address`*****`[:`*****`mask`*****`]`**	
Sets the Ethernet TAP probe IP address and optional netmask	
`address`	IP address in dotted-decimal format (e.g., 128.8.1.1). When entering the IP address by itself (without also entering the netmask), the Ethernet TAP probe uses the standard netmask assigned to that IP address.

mask	Netmask in dotted-decimal format (e.g., 255.255.0.0). If subnetting is required, you must store the netmask by entering it on the same command line, immediately following the IP address.
static_dns_server address	
Sets the DNS server to use static bootconfig	
address	IP address in dotted-decimal format (e.g. 128.1.1). The DNS server at this address will be used for domain name resolution when bootconfig is set to static.

A.2.1.2 tgtty

The `tgtty` command configures the target system serial port settings.

Syntax

```
tgtty
  [default]
  [9600 | 19200 | 38400 | 57600 | 115200]
  [data8 | data5 | data6 | data7]
  [stop1 | stop2]
  [noparity|oddparity|evenparity|lowparity|highparity]
  [<rtscts | nortscts>]
  [noxon | xon]
  [echo | noecho]
```

Options

Options can be combined in one statement. Without options, the `tgtty` command displays the current settings.

The target system serial port’s default settings are:

```
9600 data8 stop1 noparity nortscts echo
```

Default	Set the default target system serial settings
[9600 | 19200 | 38400 | 57600 | 115200]	Choose a baud rate
[data8 | data5 | data6 | data7]	Specify data bits
[stop1 | stop2]	Specify stop bits

[noparity | oddparity | evenparity | lowparity | highparity]	Define parity
[<rtscts | nortscts>]	Enable or disable hardware flow control
[noxon | xon]	Enable or disable XON/XOFF flow control
[echo | noecho]	Enable or disable target system echo feature

A.2.2 Commands to Troubleshoot Communication

The following commands are used to troubleshoot problems connecting to your network. The procedures for troubleshooting communication are covered in the Troubleshooting chapter.

NOTE In this manual, commonly used options for these commands are described.

A.2.2.1 arp

Use the `arp` command to edit the arp table by assigning *`hostnames`* to specific Ethernet addresses. Without options, it displays the current arp table.

Syntax

```
arp [-s hostname ethernet_address | -d hostname]
```

Options

`-s` *`hostname`* `ethernet_`*`address`*	Assign a *`hostname`* alias to an Ethernet address in the arp table
`-d` *`hostname`*	Delete a *`hostname`* alias from the arp table

A.2.2.2 host

Use the `host` command to edit the host table by assigning *`hostnames`* to specific IP addresses without permanently storing the routing tables in the flash EPROM of the Ethernet TAP probe. Without options, it displays the current host table.

Syntax

```
host [add hostname ip_address | delete hostname ip_address]
```

Options

`add` *`hostname ip_address`*	Assign a *`hostname`* alias to an IP address in the host table.
`delete` *`hostname ip_address`*	Delete a *`hostname`* alias from the host table.

A.2.2.3 netstat

Displays network information and statistics.

Syntax

```
netstat -a --inet | -i | -s | -r
```

Options

`-a --inet`	Displays network connections
`-i`	Displays Ethernet device status
`-s`	Displays protocol statistics
`-r`	Displays route table

A.2.2.4 ping

Use the `ping` command to verify that the Ethernet TAP probe is connected to your network.

Syntax

```
ping [-s size] [-c cnt][hostname | ip_address]
```

Options

`hostname`	Use the *`hostname`* stored in Ethernet TAP probe host table (see the host command).
`ip_address`	Use the IP address of the host you are trying to reach.
`size`	The size, in bytes, to use for request packets.
`cnt`	The number of packets to send.

A.2.2.5 route

Use the `route` command to test network routing without permanently storing the routing tables in the Ethernet TAP probe flash EPROM. Without options, it displays the current route table or default gateway.

Syntax

```
route [add destination gateway |
        delete destination]
```

Options

`add` *`destination gateway`*	Adds a dynamic route to the route table.
`delete` *`destination`*	Deletes a dynamic route from the route table.

B

Network Administration

This appendix guides the network administrators in installing a Ethernet TAP probe. The Ethernet TAP probe is an Ethernet host device that may be configured for TCP/IP using DHCP to acquire its IP configuration (the default method) or through a static IP configuration.

This appendix contains the following sections:

- CodeWarrior TAP Probe Network Ports
- Configuring the Ethernet TAP Probe Using netparam
- Using CCS to Search for CodeWarrior TAP Probes

B.1 CodeWarrior TAP Probe Network Ports

Software uses several network ports to communicate with a CodeWarrior TAP. In case the CodeWarrior TAP and host software are on the same network, you do not need to be aware of these ports. However, in case where a CodeWarrior TAP is located in a protected network, an administrator will need to provide access to these ports if you want to connect to the CodeWarrior TAP from another network. This table lists the ports used by the CodeWarrior TAP and a brief description of each port.

Table B.1 CodeWarrior TAP Network Ports

Port Number	Description
23	Telnet access to configuration console
1082	Telnet access to target serial port
1087	Used for firmware updates and by CodeWarrior to initialize the TAP
2345	Used by GDB to control the TAP
41474	Used by CodeWarrior to control the TAP

B.2 Configuring the Ethernet TAP Probe Using netparam

Use the `netparam` command to select the network parameters:

- Address resolution protocol
- Static address resolution data
- Static routing tables

CAUTION `netparam` writes its settings into non-volatile flash memory on the Ethernet TAP probe. Each time you enter a `netparam` command, wait for the `core>` prompt to reappear before entering the next command.

B.2.1 Configuring a Dynamic IP Address

To configure a dynamic IP address

1. Connect to the Ethernet TAP probe internal setup utility, as explained in the Connecting to the CodeWarrior TAP Probe Setup Utility section.
2. At the `core>` prompt, use `netparam` to specify the protocol appropriate to your network:

   ```
   netparam bootconfig dhcp[:hostname]
   ```

   DHCP is the default setting. If you specify a *`hostname`* for the CodeWarrior TAP probe, the probe will attempt to register the host name with the DHCP server, which may then update any name servers on the network.

B.2.2 Configuring a Static IP Address

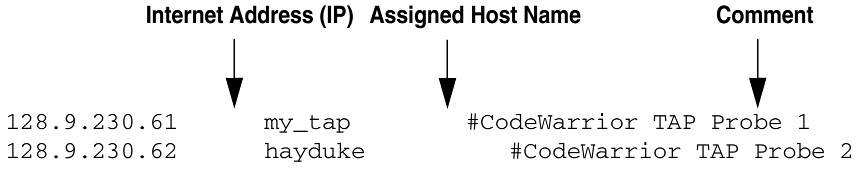
If you do not have a DHCP server on your network or you prefer to manually configure your network settings, the Ethernet TAP probe is capable of storing its IP address and netmask in flash memory. When `bootconfig` is set to `static`, the Ethernet TAP probe uses this stored information to resolve its own IP and netmask requests.

NOTE Because this is a simple proven way to add a Ethernet TAP probe to any TCP/IP network, we strongly recommend using it if you have any network communication problems.

To enter the IP and optional netmask in flash

1. Have your network administrator assign an unused IP address and host name to the probe.
2. Enter the name/address pair into the `hosts` database file. Windows `hosts` files are typically located in the `%system_root%\system32\drivers\etc\` directory.

   The following is an example of probe entries in a `hosts` file:



```
128.9.230.61      my_tap        #CodeWarrior TAP Probe 1
128.9.230.62      hayduke          #CodeWarrior TAP Probe 2
```

NOTE You should create or update the `hosts` file on the network server or on each local workstation that needs access to the probe.

3. At the `core>` prompt, use `netparam` to set and store the IP address and netmask (subnetting only) in the Ethernet TAP probe flash EPROM.

   `netparam static_ip_address` *`nnn.nnn.nnn.nnn`* `[:mmm.mmm.mmm.mmm]`

   where *`nnn.nnn.nnn.nnn`* represents the IP address and `mmm.mmm.mmm.mmm` represents the subnetting mask.

B.2.3 Static Routing

The simplest networks consist of one or more subnets. Routers forward network traffic from one point on the network to another across these subnets.

If the CodeWarrior TAP probe uses DHCP to automatically acquire its network settings, it is most likely that a default gateway setting was acquired and the probe will be accessible on other subnets.

However, when using a static IP configuration or where the DHCP configuration is incomplete, you may have to provide additional routing information, including:

- Store a default gateway in flash memory
- Load static routing tables into flash memory

B.2.3.1 Specify a default gateway or static route table (optional)

If you are using a static IP configuration or your DHCP configuration does not specify a default gateway, you can manually enter the IP address of the default gateway to use. This gateway must be accessible on your local subnet.

To specify a default gateway

A default gateway entry must specify the IP address of the first gateway that the network traffic from probe crosses. This gateway must be aware of the network's complete route table. Use the following `netparam` syntax:

```
netparam add_route 0.0.0.0 gateway_ip 1
```

For *`gateway_ip`*, provide the IP address of the router or gateway in dot notation. The default value is `0.0.0.0`.

B.2.4 Changing an Existing Route Entry

NOTE When entered in the Ethernet TAP probe, static routes are not updated automatically. You must update these routes if changes in network topology affect the static routes.

Before entering static routes, make a map of all gateway paths between the Ethernet TAP probe, as starting point, and each workstation that must have access to it.

To change an existing routing entry

1. At the `core>` prompt, delete the existing routing entry:

   ```
   netparam delete_route host_ip
   ```

2. Enter the new route as described above:

   ```
   netparam add_route host_ip gateway_ip hop_#
   ```

NOTE *`host_ip`* can identify an individual workstation or a network serving multiple hosts. The *`gateway_ip`* is the first gateway the probe traffic crosses when communicating with the destination workstation. The *`hop_#`* is the decimal number of gateways between the probe and the destination workstation.

B.2.5 Entering Static Routes

NOTE When entered in the Ethernet TAP probe, static routes are not updated automatically. You must update these routes if changes in network topology affect the static routes.

Before entering static routes, make a map of all gateway paths between the Ethernet TAP probe, as starting point, and ensure each workstation has access to it.

To enter a static route or default gateway

1. At the `core>` prompt, use the `netparam` command to enter the first host/gateway pair:

   ```
   netparam add_route host_ip gateway_ip hop_#
   ```

   Wait for the `core>` prompt between each `netparam` entry.

NOTE *`host_ip`* can identify an individual host or a network serving multiple hosts. The *`gateway_ip`* is the first gateway the Ethernet TAP probe crosses when communicating with the destination host. The *hop_#* is the decimal number of gateways between the Ethernet TAP probe and the destination host. The `netparam` command is described in the Ethernet TAP Probe Setup Utility Commands section.

2. Add routes until all destination hosts or networks are defined.
3. When the `core>` prompt returns, reset the Ethernet TAP probe by cycling power, or by entering the `reset` command.

B.2.5.1 Static route example

Figure B.1 shows three class “C” networks joined together by a single IP router, making each Ethernet TAP probe accessible from three workstations (elmer, tweety, and brutus).

No static routing information is required to make a Ethernet TAP probe accessible from a workstation local to it on a network. For example, the Ethernet TAP probe goofy on network 198.9.230.0 communicates directly with workstation elmer.

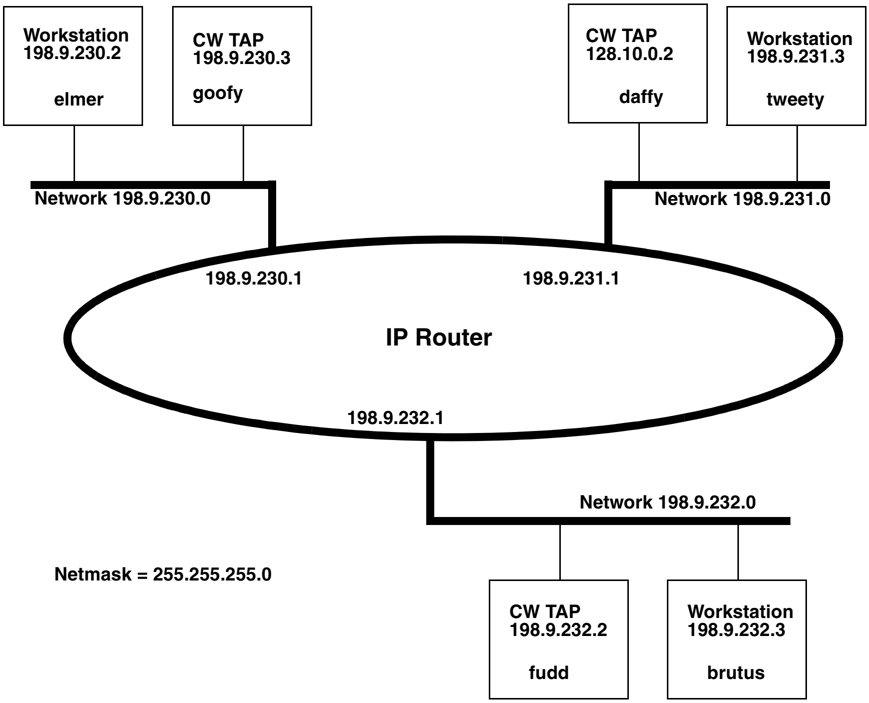
When static routing is used, a routing entry is required on a Ethernet TAP probe for each workstation on a non-local network that accesses it. The Ethernet TAP probe goofy requires two entries, for workstation tweety on network 198.9.231.0 and workstation brutus on network 198.9.232.0.

Each static route entry is made using a `netparam` command and consists of a network address and a host address. The `netparam` commands for the static route entries for Ethernet TAP probe goofy are:

```
netparam add_route 198.9.231.0 198.9.230.1 1
```

```
netparam add_route 198.9.232.0 198.9.230.1 1
```

Figure B.1 Three Class C Networks Connected by a Single Router



Each of the three Ethernet TAP probe hosts must have a static route entry for each remote workstation that accesses it.

B.3 Using CCS to Search for CodeWarrior TAP Probes

The CCS console provides a `findcc` command line utility which searches for all the CodeWarrior probes on the local subnet of each network interface of the host and lists the probes' IP addresses, as well as other information. It uses mDNS multicast packets to discover CodeWarrior TAP devices. Multicast packets are typically limited to the local subnet and typically are not routed or passed through VPNs.

If your CodeWarrior TAP probe acquires its IP address using DHCP, but is not able to register its host name on the network, you will need the probe's IP address. To find the probe's address, perform the following steps:

1. Launch CCS and open the CCS Command window. The procedure is slightly different on Windows and Linux/Solaris host machines.
   - For Windows, run the command:

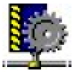
     ```
     ccs\bin\ccs.exe
     ```

     This will launch CCS and add a CCS icon (see Figure B.2) to your taskbar. Double-click that icon in the taskbar to open the Command window.
   - For Linux/Solaris, run the command:

     ```
     ccs/bin/ccs
     ```

     This will launch CCS and open the Command window automatically.

Figure B.2 CCS Icon



2. The `findcc` command takes the following arguments when searching for CodeWarrior TAP probes:

   ```
   findcc cwtaps [-quiet|-verbose]
   ```

B.3.0.1 Sample output

```
% findcc cwtaps
FSL021351 (192.168.0.145): CodeWarrior TAP
  Power Architecture JTAG/COP Probe Tip
  Boot Loader v0.9.2
  Operating System v0.9.8
%
```

C

Ethernet TAP Probe Firmware (Core)

This appendix explains the methods for reprogramming the Boot Loader and Operating System images stored in the flash EPROM of the Ethernet TAP probe. Before reprogramming the flash EPROM, make sure you have already configured the Ethernet TAP probe network communication.

This appendix contains the following sections:

- CodeWarrior TAP Probe Internal Software Overview
- Reprogramming the Ethernet TAP Probe Firmware Images

C.1 CodeWarrior TAP Probe Internal Software Overview

C.1.1 Boot Loader

The CodeWarrior TAP Boot Loader image performs hardware initialization and starts up the OS. When the CodeWarrior TAP first powers up, it executes the Boot Loader. This occurs while the heartbeat LED is solid red. The Boot Loader is not generally visible to the user and should rarely require reprogramming or updating. If an update is required, follow the instructions in the Reprogramming the Firmware through the Ethernet Port section.

C.1.2 Operating System

The Ethernet TAP probe OS image provides tools for configuring and testing network communication, for re-loading the probe software and the underlying software framework required to work with the debugger.

When the Ethernet TAP probe finishes executing the Boot Loader, it loads the OS. This is indicated by the `core>` prompt in the Ethernet TAP probe's setup utility, and by flashing a orange or green heartbeat LED. To reprogram the OS image stored in the Ethernet TAP probe flash EPROM, see the Reprogramming the Ethernet TAP Probe Firmware Images section.

C.1.3 Shell Software

The Ethernet TAP probe shell software is transparent to the user, and the application tells the probe how to control the target system. It recognizes the specific target system processor and debug port interface, and carries out the instructions of the debugger. The shell software is automatically stored and updated in flash, and therefore does not require manual reprogramming.

C.2 Reprogramming the Ethernet TAP Probe Firmware Images

At some point you may be required to reprogram the Ethernet TAP probe firmware images 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 Boot Loader or Operating System software. The firmware is distributed in two images:

- `cwtap_bl.gp` contains the Boot Loader
- `cwtap_os.gp` contains the Operating System

A flash file loader (`updatecwtap`) utility is included with the debugger software. `updatecwtap` provides the ability to reprogram the Ethernet TAP probe firmware images stored in its flash EPROM.

C.2.1 Reprogramming the Firmware through the Ethernet Port

In order to use the following instructions, the Ethernet TAP probe communications must already be configured (refer to the Connecting to a Network chapter or Setting Up a Standalone PC Ethernet appendix).

To reprogram the firmware image

1. Launch CCS and open the CCS command window. For information on launching CCS, refer to the Using CCS to Search for CodeWarrior TAP Probes topic.
2. In the CCS Command window, enter the command:

   ```
   updatecwtap {hostname | ip_address}
   ```
3. As it executes, `updatecwtap` reports its progress. When the process is complete, `updatecwtap` reports:

   ```
   All updates completed successfully.
   ```

CAUTION Do nothing to disrupt operation while running the `UPDATECWTAP` command. The heartbeat LED will flash at a faster frequency while the update is in progress, and the probe will automatically reboot when the update is complete. Power failures, network disruptions, and Ethernet TAP probe resets during an update and can create a non-working state that may require factory repair.

These procedures must be performed on each Ethernet TAP probe that you plan to use with the current version of debugger.

JTAG/COP Connector Information

The CodeWarrior TAP JTAG/COP probe has a 16-pin connector which automatically supports target system signal levels from 1.2V to 3.3V.

Figure D.1 shows the pin assignments of the probe JTAG/COP connector.

Table D.1 lists JTAG/COP signal names, direction, pin numbers, descriptions, and drive capabilities for the probe JTAG/COP connector.

Table D.2 provides a general description of each JTAG/COP signal and the operational requirements.

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

Figure D.1 Ethernet TAP Probe for JTAG/COP Connector Pin Assignments

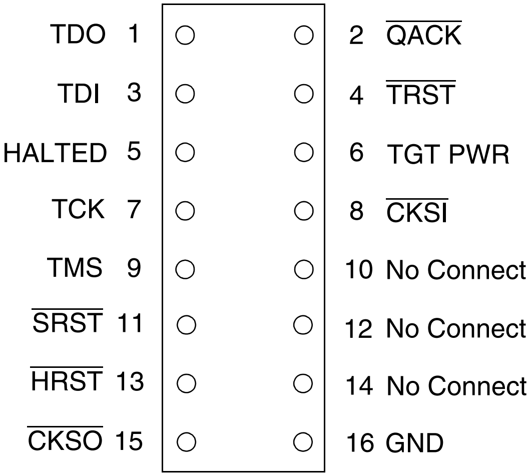

TDO 1
2 $\overline{\text{QACK}}$
TDI 3
4 $\overline{\text{TRST}}$
HALTED 5
6 TGT PWR
TCK 7
8 $\overline{\text{CKSI}}$
TMS 9
10 No Connect
$\overline{\text{SRST}}$ 11
12 No Connect
$\overline{\text{HRST}}$ 13
14 No Connect
$\overline{\text{CKSO}}$ 15
16 GND

Table D.1 Ethernet TAP Probe for JTAG/COP Signal Directions

JTAG/ COP Pin	Signal Mnemonic	Signal Direction	Description
1	TDO	From target system	17pF load
2	$\overline{\text{QACK}}$	From Ethernet TAP probe connector	100Ω pull-down
3	TDI	From Ethernet TAP probe connector	50mA driver
4	$\overline{\text{TRST}}$	From Ethernet TAP probe connector	50mA driver
5	HALTED	From target system	17pF load
6	TGT PWR	From target system	2MΩ pull-down, plus 0.01µF load
7	TCK	From Ethernet TAP probe connector	50mA driver
8	CKSI	From Ethernet TAP probe connector	50mA driver
9	TMS	From Ethernet TAP probe connector	50mA driver
10	No Connect	- n/a -	
11	$\overline{\text{SRST}}$	Bi-directional	Open-drain. 5Ω to ground when asserted by Ethernet TAP probe, 22pF load when not asserted[1]
12	No Connect	- n/a -	

Table D.1 Ethernet TAP Probe for JTAG/COP Signal Directions (*continued*)

JTAG/ COP Pin	Signal Mnemonic	Signal Direction	Description
13	$\overline{\text{HRST}}$	Bi-directional	Open-drain. 5Ohm to ground when asserted by Ethernet TAP probe, 22pF load when not asserted[1]
14	No Connect	- n/a -	
15	$\overline{\text{CKSO}}$	From target system	17pF load
16	GND	- n/a -	

[1] 4.7KΩ pull-up to buffered TGT PWR.

Table D.2 Ethernet TAP Probe for JTAG/COP Signal Recommendations and Requirements

JTAG/ COP Pin	Signal Mnemonic	Requirement
1	TDO	Must be wired to the target system processor. TDO is an output from the target system processor and an input to the Ethernet TAP probe. The TDO trace run should be kept short, and should maintain a "two-signal-width" spacing from any other parallel dynamic signal trace. TDO should have a series termination resistor located near the target system processor.
2	$\overline{\text{QACK}}$	May be wired to the target system processor. $\overline{\text{QACK}}$ is an input to most PowerPC processors and must remain low while the Ethernet TAP probe is connected to the target system. The Ethernet TAP probe connects this signal internally to the JTAG/COP GND pin (16) through a 100Ω resistor.
3	TDI	Must be wired to the target system processor. The Ethernet TAP probe drives the TDI output with up to 50mA. The TDI trace should be kept short, and should maintain a "two-signal-width" spacing from any other parallel dynamic signal trace. TDI should have an RC termination option at the processor.

Table D.2 Ethernet TAP Probe for JTAG/COP Signal Recommendations and Requirements (*continued*)

JTAG/ COP Pin	Signal Mnemonic	Requirement
4	$\overline{\text{TRST}}$	Must be wired to the target system processor. The Ethernet TAP probe drives the $\overline{\text{TRST}}$ output with up to 50mA. To gain control of the processor, the Ethernet TAP probe negates $\overline{\text{TRST}}$ approximately 250ms before negation of $\overline{\text{HRST}}$. This allows the Ethernet 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 should maintain a "two-signal-width" spacing from any other parallel dynamic signal trace.
5	HALTED	Need not be wired to the target system. The Ethernet TAP probe does not currently use this signal.
6	TGT PWR	Must be wired to the target system. The Ethernet TAP probe uses this signal to determine if power is applied to the target system. This signal is also used as a voltage reference for the signals driven by the Ethernet TAP probe ($\overline{\text{CKSI}}$, $\overline{\text{TRST}}$, TCK, TMS, TDI). TGT PWR (pin 6) should be connected to the target system Vcc through a pull-up resistor. The CodeWarrior TAP will draw less than 50 µA from this signal, so a weak pull-up is sufficient (1KOhm).
7	TCK	Must be wired to the target system processor. The Ethernet TAP probe drives the TCK output with up to 50mA. The TCK trace run should be kept as short as possible, and should maintain a "two-signal-width" spacing from any other parallel dynamic signal trace.
8	CKSI	Need not be wired to the target system. The Ethernet TAP probe does not currently use this signal.
9	TMS	Must be wired to the target system processor. The Ethernet TAP probe 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

Table D.2 Ethernet TAP Probe for JTAG/COP Signal Recommendations and Requirements (*continued*)

JTAG/ COP Pin	Signal Mnemonic	Requirement
11	$\overline{\text{SRST}}$	May be wired to the target system processor. During reset, the Ethernet TAP probe drives $\overline{\text{SRST}}$ to ground through a 5 Ohm resistor.
12	No Connect	Not required for emulation
13	$\overline{\text{HRST}}$	Must be wired to the target system processor. During reset, the Ethernet TAP probe drives $\overline{\text{HRST}}$ to ground through a 5 Ohm resistor.
14	No Connect	Not required for emulation
15	$\overline{\text{CKSO}}$	Should be wired to the target system processor. The Ethernet TAP probe senses $\overline{\text{CKSO}}$ to determine if the processor halted execution in a checkstop state.
16	GND	Must be wired to the target system. GND is connected directly to the ground inside the Ethernet TAP probe.

E

OnCE Connector Information

The CodeWarrior TAP OnCE probe has a 14-pin connector that automatically supports target system signal levels from 1.2V to 3.3V.

Figure E.1 shows the pin assignments of the probe OnCE connector.

Table E.1 lists OnCE signal names, direction, pin numbers, descriptions, and drive capabilities for the probe OnCE connector.

Table E.2 provides a general description of each OnCE signal and the operational requirements.

NOTE All OnCE signals must meet accepted standards for OnCE signal design. To ensure proper and stable operation between the Ethernet TAP probe and the target system, the OnCE signals must meet the requirements listed in Table E.2.

Figure E.1 Ethernet TAP Probe OnCE Connector Pin Assignments

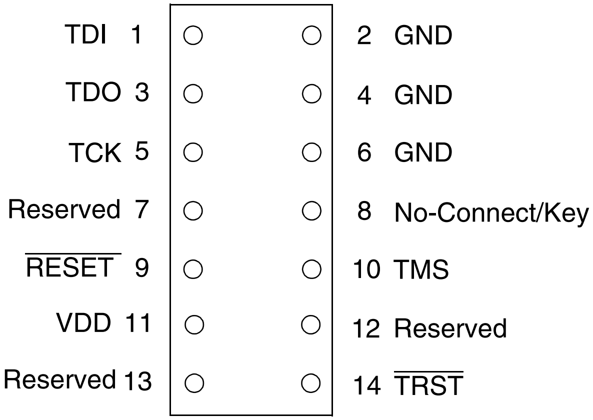


Table E.1 Ethernet TAP Probe OnCE Signal Directions

OnCE Pin	Signal Mnemonic	Signal Direction	Description
1	TDI	From Ethernet TAP probe connector	50mA driver
2	GND	- n/a -	
3	TDO	From target system	17pF load
4	GND	- n/a -	
5	TCK	From Ethernet TAP probe connector	50mA driver
6	GND	- n/a -	
7	Reserved	From Ethernet TAP probe connector	50mA driver
8	No-Connect/ Key	- n/a -	
9	$\overline{\text{RESET}}$	Bi-directional	Open-drain. 5Ω to ground when asserted by Ethernet TAP probe, 22pF load when not asserted[1]
10	TMS	From Ethernet TAP probe connector	50mA driver
11	VDD	From target system	2MΩ pull-down, plus 0.01uF load
12	Reserved	Bi-directional	Open-drain, 5Ω to ground when asserted by Ethernet TAP probe, 22pF load when not asserted[1]

Table E.1 Ethernet TAP Probe OnCE Signal Directions (*continued*)

OnCE Pin	Signal Mnemonic	Signal Direction	Description
13	Reserved	From target system	17pF load
14	$\overline{\text{TRST}}$	From Ethernet TAP probe connector	50mA driver

[1] 4.7KΩ pull-up to buffered VDD.

Table E.2 Ethernet TAP Probe OnCE Signal Recommendations and Requirements

OnCE Pin	Signal Mnemonic	Requirement
1	TDI	Must be wired to the target system processor. The Ethernet 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.
2	GND	Must be wired to the target system. GND is connected directly to the ground inside the Ethernet TAP probe.
3	TDO	Must be wired to the target system processor. TDO is an output from the target system processor and input to the Ethernet TAP probe. The TDO trace run should be kept short, and should maintain a "two-signal-width" spacing from any other parallel dynamic signal trace. TDO should have a series termination resistor located near the target system processor.
4	GND	Must be wired to the target system. GND is connected directly to the ground inside the Ethernet TAP probe.
5	TCK	Must be wired to the target system processor. The Ethernet TAP probe drives the TCK output with up to 50mA. The TCK trace run should be kept as short as possible, and should maintain a “two-signal-width” spacing from any other parallel dynamic signal trace.
6	GND	Must be wired to the target system. GND is connected directly to the ground inside the Ethernet TAP probe.

Table E.2 Ethernet TAP Probe OnCE Signal Recommendations and Requirements

OnCE Pin	Signal Mnemonic	Requirement
7	Reserved	Not required for emulation.
8	No-Connect/ Key	Not required for emulation. Pin 8 should be clipped on the target system OnCE header.
9	$\overline{\text{RESET}}$	Must be wired to the target system processor. During reset, the Ethernet TAP probe drives $\overline{\text{RESET}}$ to ground through a 5Ω resistor.
10	TMS	Must be wired to the target system processor. The Ethernet TAP probe drives the TCK output with up to 50mA. The TCK trace run should be kept as short as possible, and should maintain a “two-signal-width” spacing from any other parallel dynamic signal trace.
11	VDD	Must be wired to the target system. The Ethernet TAP probe uses this signal to determine if power is applied to the target system. This signal is also used as a voltage reference for the signals driven by the Ethernet 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 system processor. The Ethernet TAP probe drives the $\overline{\text{TRST}}$ output with up to 50 mA. The $\overline{\text{TRST}}$ trace run should be kept short, and should maintain a “two-signal-width” spacing from any other parallel dynamic signal trace.

ColdFire BDM Connector Information

The CodeWarrior TAP ColdFire BDM probe has a 26-pin connector that automatically supports target signal levels from 1.2V to 3.3V.

Figure F.1 shows the pin assignments of the probe BDM connector.

Table F.1 lists BDM signal names, direction, pin numbers, descriptions, and drive capabilities for the probe BDM connector.

Table F.2 provides a general description of each BDM signal and the operational requirements.

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 F.2.

ColdFire BDM Connector Information

Figure F.1 CodeWarrior TAP Probe for ColdFire BDM Connector Pin Assignments

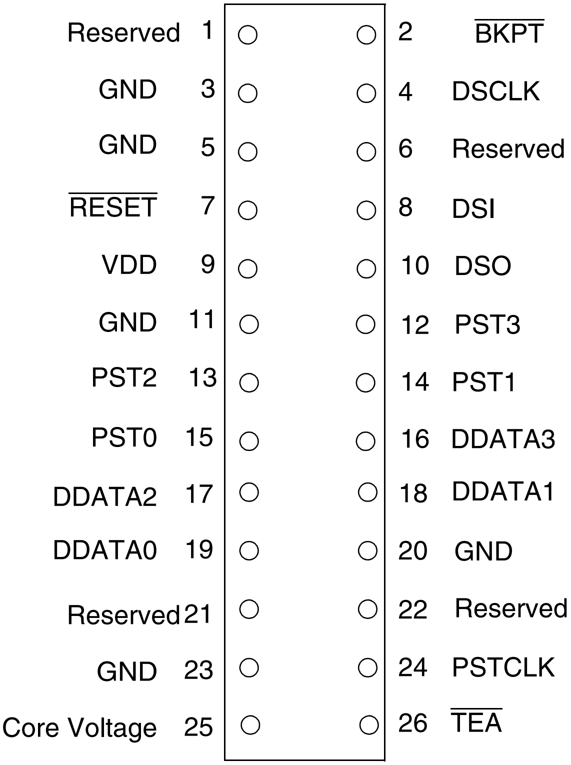


Table F.1 CodeWarrior TAP Probe for ColdFire BDM Signal Directions

BDM Pin	Signal Mnemonic	Signal Direction	Description
1	Reserved	-n/a-	
2	BKPT	Bi-directional	Open-drain. 5 Ω to ground when asserted by USB TAP probe, 22pF load when not asserted[1]
3	GND	- n/a -	
4	DSCLK	From CodeWarrior TAP probe connector	50mA driver

Table F.1 CodeWarrior TAP Probe for ColdFire BDM Signal Directions (*continued*)

BDM Pin	Signal Mnemonic	Signal Direction	Description
5	GND	- n/a -	
6	Reserved	- n/a -	
7	$\overline{\text{RESET}}$	Bi-directional	Open-drain. 5Ω to ground when asserted by USB TAP probe, 22pF load when not asserted[1]
8	DSI	From CodeWarrior TAP probe connector	50mA driver
9	VDD	From target system	2MΩ pull-down, plus 0.01uF load
10	DSO	From target system	22pF load
11	GND	- n/a -	
12	PST3	CodeWarrior	17pF load
13	PST2	CodeWarrior	17pF load
14	PST1	CodeWarrior	17pF load
15	PST0	CodeWarrior	17pF load
16	DDATA3	CodeWarrior	17pF load
17	DDATA2	CodeWarrior	17pF load
18	DDATA1	CodeWarrior	17pF load
19	DDATA0	CodeWarrior	17pF load
20	GND	- n/a -	
21	Reserved	- n/a -	
22	Reserved	- n/a -	
23	GND	- n/a -	
24	PSTCLK	CodeWarrior	17pF load

Table F.1 CodeWarrior TAP Probe for ColdFire BDM Signal Directions (*continued*)

BDM Pin	Signal Mnemonic	Signal Direction	Description
25	Core Voltage	- n/a -	
26	$\overline{\text{TEA}}$	- n/a -	

[1] 4.7 Ω pull-up to buffered TGT PWR.

Table F.2 CodeWarrior TAP Probe for ColdFire BDM Signal Recommendations and Requirements

BDM Pin	Signal Mnemonic	Requirement
1	Reserved	Need not be wired to the target. The CodeWarrior 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 CodeWarrior TAP probe.
4	DSCLK	DSCLK must be connected to the target's processor. It is driven by the CodeWarrior TAP probe as an output with up to 50mA. This signal is the clock for the BDM interface. It is a good design practice to keep the trace length short and isolate the trace from other signals. If the trace is long, then termination may be needed.
5	GND	Must be wired to the target. GND is connected directly to the CodeWarrior TAP ground inside the CodeWarrior TAP probe.
6	Reserved	Need not be wired to the target. The CodeWarrior TAP probe does not currently use this signal.
7	$\overline{\text{RESET}}$	Must be wired to the target. During reset, the CodeWarrior TAP probe drives $\overline{\text{RESET}}$ to ground through a 100 Ω resistor.
8	DSI	Must be wired to the target processor. The CodeWarrior TAP probe drives the TDI output with up to 50mA.

Table F.2 CodeWarrior TAP Probe for ColdFire BDM Signal Recommendations and Requirements

BDM Pin	Signal Mnemonic	Requirement
9	VDD	Must be wired to the target. The CodeWarrior 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 CodeWarrior 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 CodeWarrior TAP probe.
11	GND	Must be wired to the target. GND is connected directly to the ground inside the CodeWarrior TAP probe.
12	PST3	Need not be wired to the target. The CodeWarrior TAP probe uses this signal for trace collection.
13	PST2	Need not be wired to the target. The CodeWarrior TAP probe uses this signal for trace collection.
14	PST1	Need not be wired to the target. The CodeWarrior TAP probe uses this signal for trace collection.
15	PST0	Need not be wired to the target. The CodeWarrior TAP probe uses this signal for trace collection.
16	DDATA3	Need not be wired to the target. The CodeWarrior TAP probe uses this signal for trace collection.
17	DDATA2	Need not be wired to the target. The CodeWarrior TAP probe uses this signal for trace collection.
18	DDATA1	Need not be wired to the target. The CodeWarrior TAP probe uses this signal for trace collection.
19	DDATA0	Need not be wired to the target. The CodeWarrior TAP probe uses this signal for trace collection.
20	GND	Must be wired to the target. GND is connected directly to the ground inside the CodeWarrior TAP probe.
21	Reserved	Need not be wired to the target. The CodeWarrior TAP probe does not currently use this signal.
22	Reserved	Need not be wired to the target. The CodeWarrior TAP probe does not currently use this signal.

Table F.2 CodeWarrior TAP Probe for ColdFire BDM Signal Recommendations and Requirements

BDM Pin	Signal Mnemonic	Requirement
23	GND	Must be wired to the target. GND is connected directly to the ground inside the CodeWarrior TAP probe.
24	PSTCLK	May be wired to the target. The CodeWarrior TAP probe uses this signal to support synchronous clocking mode.
25	Core Voltage	Need not be wired to the target. The CodeWarrior TAP probe does not currently use this signal.
26	$\overline{\text{TEA}}$	Need not be wired to the target. The CodeWarrior TAP probe does not currently use this signal.

G

Troubleshooting

This appendix provides Ethernet TAP probe troubleshooting information.

This appendix contains the following sections:

- Troubleshooting Communications Problems
- Troubleshooting Power Problems

G.1 Troubleshooting Communications Problems

This section explains how to troubleshoot communication problems between the debugger and the Ethernet TAP probe.

If the debugger is unable to communicate with the Ethernet TAP probe:

- Check the cable and connections between the network cable and the Ethernet TAP probe.

  The Ethernet TAP probe connects directly to networks that use twisted pair (10/100BaseT) cables.
- Make sure communication was configured correctly for your network.
- Make sure the Ethernet TAP probe is receiving power. Refer to the CodeWarrior TAP Probe Status Indicators section for a description of the status LEDs.
- Make sure the Ethernet TAP probe is running the OS software. For more information on loading the OS software, refer to the Ethernet TAP Probe Firmware (Core) topic.
- Use the communication troubleshooting utilities of Ethernet TAP probe to verify that it is recognized on your network, or to help diagnose problems connecting to your network.

  To troubleshoot communication, refer to the Verify Network Communication topic. To list all the CodeWarrior probes on your local subnets, use the CCS `findcc` host utility. Refer to the Using CCS to Search for CodeWarrior TAP Probes topic.
- Make sure the debugger is set up correctly for Ethernet communication with the Ethernet TAP probe.

  If all the settings are correct and the debugger cannot communicate with the Ethernet TAP probe, contact the Customer Support for assistance.

G.1.1 Verify Network Communication

If you want to verify that the CodeWarrior TAP is up and running on your network, enter the `ping` command at the `core>` prompt of the Ethernet TAP probe.

To verify network communication

1. Connect to the Ethernet TAP probe internal setup utility, as explained in the Connecting to the CodeWarrior TAP Probe Setup Utility section.
2. Verify communication by entering this command at the `core>` prompt:

   ```
   ping ipaddress | hostname
   ```

   For example, to ping a hostname, named `my_tap` at IP address `128.9.230.61`, enter the command as follows:

   ```
   ping 128.9.230.61
   ```

   — or —

   ```
   ping my_tap
   ```

NOTE When establishing communication, you will have to ping the IP address that was used during the setup process, as the Ethernet TAP probe may not automatically recognize the hostname. To ping a hostname, the Ethernet TAP probe internal host table must first be updated

G.1.2 View Network Connections

If you want to check your network configuration and activity, use the `netstat` command. This command displays all the network statistics on active connections such as their current status, all hosts that are connected, and which programs are running. You can also see information about the routing table and even get statistics on your network interfaces.

To run the `netstat` command

1. Connect to the internal setup utility of Ethernet TAP probe.
2. At the `core>` prompt, enter the `netstat` command using this syntax:

   ```
   netstat -s
   ```

   The output of this command indicates whether any data is being sent or received over the network. For description of the netstat options, see the netstat section.

G.2 Troubleshooting Power Problems

If the Ethernet TAP probe behaves erratically, check the connections to the external power supply.

The LED labeled HEARTBEAT indicates whether the Ethernet TAP probe is receiving power. If this LED is not lit, check the connections to the external power supply.

Index

A
AC characteristics 32
ARP command 40

B
BDM connector information 65

C
Cables
    RJ-45 12
CodeWarrior
    communications 11
    gateways 47
    setting up CodeWarrior communications 13
    subnets 47
CodeWarrior address 13
CodeWarrior TAP
    internal setup utility commands 35
    target connection 17
    updating 52
Commands
    troubleshooting 40
    utility 36
Communications
    accessing serial port 22
    CodeWarrior 11
    configuring Serial Port 20
    RS-232C problems 71
    telnet to CodeWarrior TAP 36
Configuration
    netparam command 44
Connector information
    BDM 65
Connector information, target
    JTAG/COP 55
    OnCE 61
Connectors
    debug port 31
    locations of 29
    RJ-25 Target Serial 30
Core software
    overview 51
    updating 52

D
Debug port
    about 17
    connector 31
DHCP 11, 38, 43
Dimensions 33
Dynamic routing 45

E
Electrical characteristics 32
Electrical requirements 8
evaluation board 17

F
findcc utility 48
finding IP address 48
Firmware
    about 51
    updating the CodeWarrior TAP's flash EPROM 52
Flash EPROM
    updating the firmware 52

G
Gateways 47

H
Hardware
    specifications 27
Highlights
    product highlights 6
HOST command 40
Host Connectors 29
    RJ-45 Ethernet 30
    USB 30
Hostname
    CodeWarrior TAP 13

I

Indicators
- RJ-45 Ethernet Connector 29
- Run/Pause 28

Internal setup utility
- CodeWarrior connection 36
- connection methods 35
- list of commands 36

IP address 13
- identifying 48

J

JTAG/COP connector information 55

L

LED Indicators 27

LEDs 27
- location of 29

N

NETPARAM command 36, 44

netparam command 13

NETSTAT command 41

Network
- routing 13
- setup 11

Network administration 43

Network routing 47

O

OnCE connector information 61

P

Physical dimensions 33

Physical fit 32

PING command 41

Power
- problems, troubleshooting 72
- source 8

probe
- IP address 48

Probe tip
- connecting to target 17

Problems
- communications 71
- fitting in target 32
- power 72

Procedures
- connecting to the target 17

Product highlights 6

Q

QACK* 10

R

Requirements
- electrical 8
- target 10

RIP 47

RJ-25 Target Serial connector 30

ROUTE command 42

Routing
- examples 47
- static table creation 47

Run/Pause indicator 28

S

Serial
- RJ-25 Target Serial connector 30
- target access 22

Setting up CodeWarrior communications 13

Setup utility
- CodeWarrior connection 36
- connection methods 35
- list of commands 36

Shell software
- overview 52

Specifications
- AC 32
- hardware 27

Starting CodeWarrior TAP System 23

Static IP 11, 38, 43

Static routing 45

Subnets 47

Support 2

T

Target connection 17
Target Connectors 30
  RJ-25 Target Serial 30
Target requirements 10
Target system control 7
TCP/IP 11
Technical support 2
Telnet to CodeWarriorTAP 36
Temperature range 33
TGTTY command 36, 39
Troubleshooting
  commands 40
  communications problems 71
  power problems 72

U

Updating
  core software 52
Utility commands 36

W

Web address 2