# CodeWarrior TAP Probe User Guide

# Contents

# Chapter 1
# Introducing 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.

---
**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.

---

This chapter explains:

- What is CodeWarrior TAP probe? on page 4
- Operating requirements on page 6
- Related documentation on page 9

## 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, 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.    CodeWarrior TAP probe**

# 1.1.1  Product highlights

This section lists the CodeWarrior TAP probe features.

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  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, T-series, Qonverge, but not PQI

- BDM for ColdFire targets (not ColdFire v1)

- OnCE for StarCore

For information on connecting to a target, see Connecting to target system topic.

## 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 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 1Mohm resistor connected to ground.

- On the work surface and floor, use static conductive mats with a 1Mohm 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

This section contains the operating temprature considerations for the CodeWarrior TAP probe.

For operating temperature of the TAP probe, see 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, see 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 power supply cable

This section explains how to connect the USB cable to the connector on the CodeWarrior TAP probe.

Connect the power supply 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.
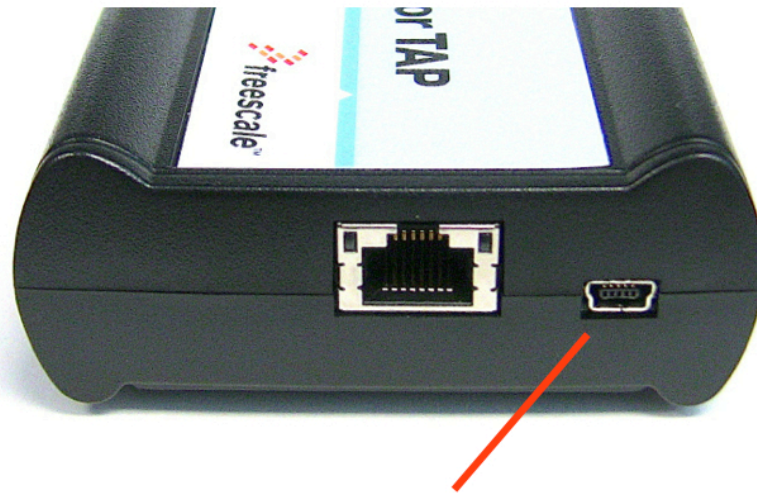
Host USB Connector and Power

**Figure 2.  CodeWarrior TAP probe with USB connector**

## 1.2.3.2  Cycling power to 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 QACK_B signal, for the CodeWarrior TAP probe to properly
> stop and restart the target, the QACK_B 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.

# Chapter 2
# Connecting to Network

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

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

This chapter explains:

- Connecting CodeWarrior TAP probe to network

- Customizing CodeWarrior TAP probe

- Testing network communication

## 2.1 Connecting CodeWarrior TAP probe to 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 FSLXXYYZZ where XXYYZZ is the last three octets of the MAC address, provided on a label on the bottom side of the probe. For example, if the probe's MAC address is 00:04:9f:00:77:31, the host name will be FSL007731. CodeWarrior TAP probe - bottom view 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.

Host Ethernet Connector
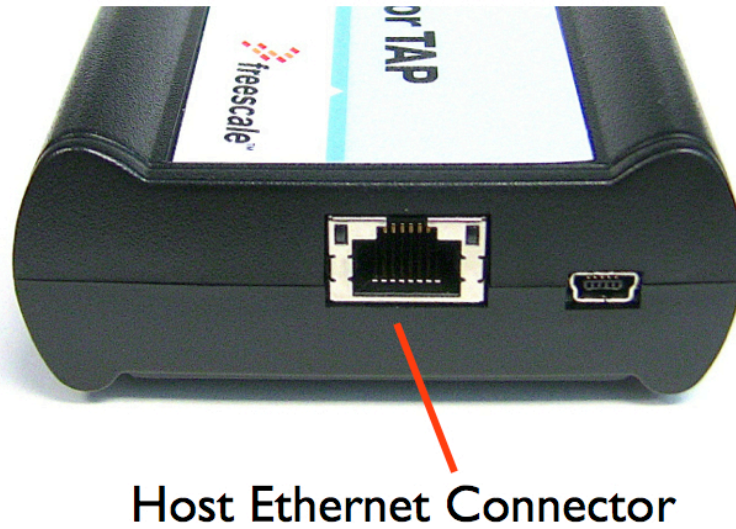
**Figure 3.     CodeWarrior TAP probe with Ethernet connector**

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

**Figure 4.     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, see Using CCS to search for CodeWarrior TAP probe 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, see Network administration topic.

---

## 2.2  Customizing CodeWarrior TAP probe

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 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 CodeWarrior TAP configuration console:

1. Connect the other end of the USB cable to the USB connector of the CodeWarrior TAP probe.



**Figure 5.     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 netparam topic for syntax and options. For more information on installing the CodeWarrior TAP probe on a network, see Network administration topic.

   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

The ping command is used to test the network communication.

You can use the `ping` command to ensure that the CodeWarrior TAP probe can communicate with the host. To verify communication, type the following at a host command prompt:

```
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.

# Chapter 3
# Connecting to 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 target system
- Connecting to 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, ColdFire BDM, or StarCore debug interfaces. The following topics describe the debug port connector specifications:

- JTAG/COP connector information
- ColdFire BDM connector information

## 3.2  Connecting to target system

The target system must have a debug port header that you can connect to the CodeWarrior TAP probe.

Make sure that you properly align the TAP multi-pin socket connector with the multi-pin header on your 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.

Connecting Probe Tip to the Probe
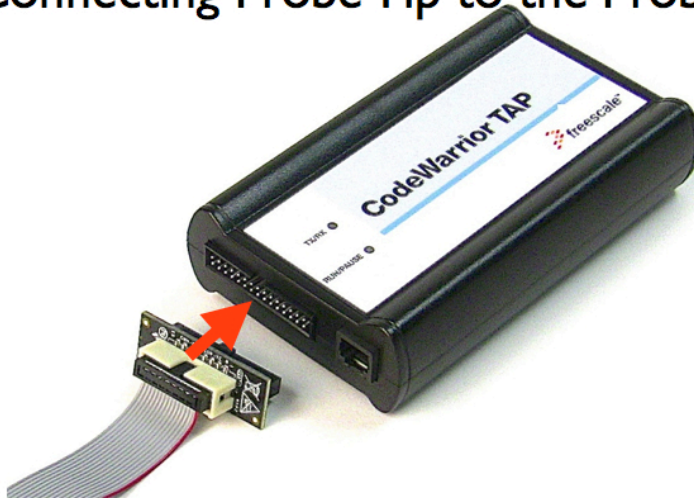
Push Probe Tip onto Probe

**Figure 6. 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 target

This section explains how to connect the CodeWarrior TAP cable to the target debug port header.

To connect the CodeWarrior 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.
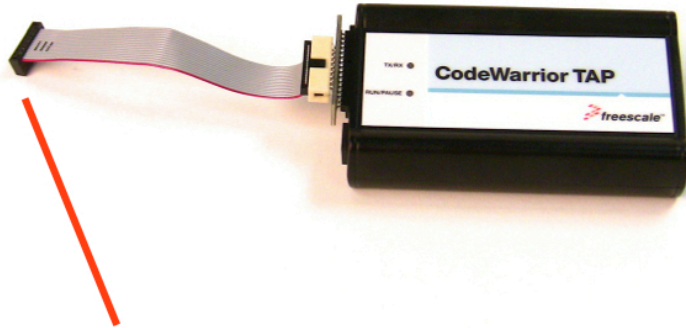
**Figure 7.     CW TAP probe - connecting to target**

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

## 3.3  Connecting to 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 8.  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 topics explain how to access the serial port of the target system:

- Connecting CodeWarrior TAP probe to target system

- Configuring target serial port

- Accessing target serial port

### 3.3.1  Connecting CodeWarrior TAP probe to 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 target serial port

If the CodeWarrior TAP probe target serial port settings do not match the serial port settings of your target system, configure CodeWarrior TAP probe serial port.

This table shows the default settings of the CodeWarrior TAP target serial port.

**Table 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 |

To configure CodeWarrior TAP probe serial port:

1. Make sure network communications are configured correctly. For more information, see Connecting to network topic.

2. Connect to the CodeWarrior TAP probe internal setup utility. For more information, see Connecting to 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 target serial port to the default settings:

1. Make sure network communications are configured correctly. For more information, see Connecting to network topic.

2. Connect to the DCU probe internal setup utility. For more information, see Connecting to 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 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, see Connecting CodeWarrior TAP probe to target system topic).

2. Verify the serial port settings.

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; see Connecting to network topic. To identify the IP address of any probe on the subnet. 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 was connected directly to the serial port of your target system.

# Chapter 4
# Using CodeWarrior TAP Probe

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

This chapter contains the following topics:

- Debugging with CodeWarrior TAP system
- Accessing CodeWarrior TAP probe remotely

## 4.1 Debugging with CodeWarrior TAP system

Before starting debugging with CodeWarrior TAP, you need to install CodeWarrior on your machine.

This topic 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 configure 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, see LEDs on CodeWarrior TAP probe topic.

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

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

The following topics provide information specific to TAP probe operation:

- Run/Pause/Mixed mode states
- Connecting to multiple TAP probes

### 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 the states: run, pause, or mixed mode.

The three states (modes) are run, pause, or mixed mode. The 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 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 9.    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 CodeWarrior TAP Probe Remotely

You can remotely access the internal setup utility and the target serial port of the probe after you connect the probe to your network.

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

• Reconfigure communications

• Use the serial port of your target system

• Reset the probe through your Ethernet connection

To remotely access the setup utility:

Open a telnet session and connect to the 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; see Connecting to network topic. To identify the IP address of any probe on the subnet, see 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 probe Target Serial port is physically connected to your target's serial port, and it is configured correctly. For more information, see Accessing target serial port topic.

# Chapter 5
# Hardware Specifications

This chapter provides hardware specifications for the probe.

The following sections are covered in this chapter:

- LEDs on CodeWarrior TAP probe

- Host connectors on CodeWarrior TAP probe

- Target connectors on CodeWarrior TAP probe

## 5.1  LEDs on CodeWarrior TAP probe

This section lists the LEDs of the CodeWarrior TAP probe.

The figure below shows the various LEDs of the CodeWarrior TAP probe.
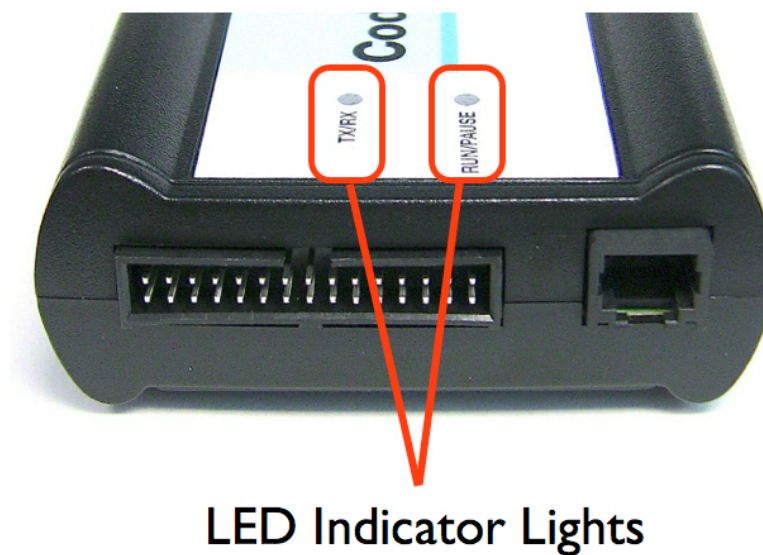


**Figure 10.  CW TAP probe - LED indicators**
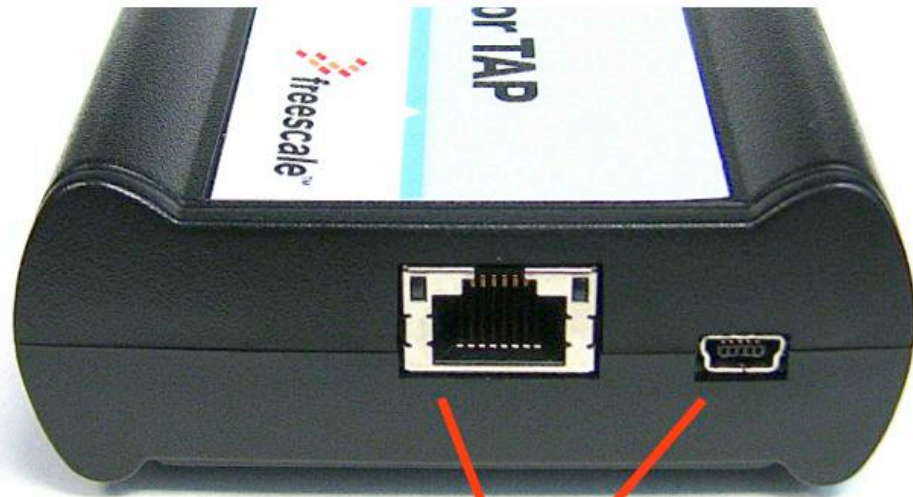
**Figure 11. CW TAP probe - Ethernet and USB connector**

## 5.1.1 Transmit/Receive indicator

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

- The LED is red until the 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 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 probe interface consists of an that connects directly to 10/100 twisted pair networks. See Connecting to network topic 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

The figure below shows the host connectors of the CodeWarrior TAP probe.
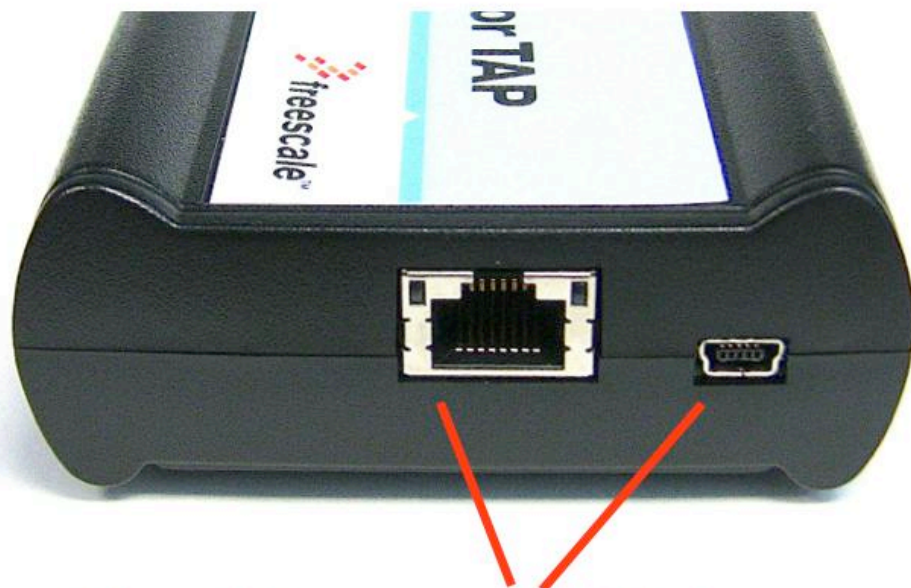


**Figure 12.  CW TAP probe - host side view**

## 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

The figure below shows the target connectors of the CodeWarrior TAP probe.
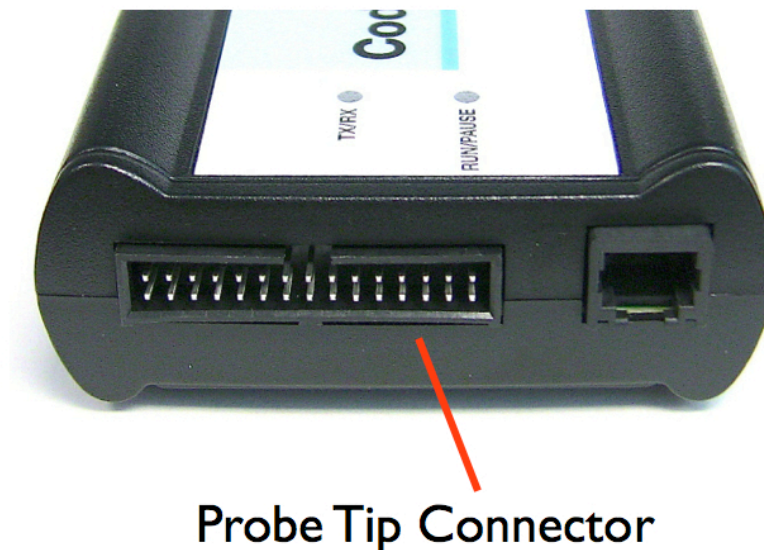


**Figure 13.  CW TAP probe - target side view**

## 5.3.1  RJ25 target serial connector

The 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 14.  CW TAP probe - target serial connector**

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

**Table 2.  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 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 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 .

## 5.3.3  Electrical characteristics

The 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 probe to the target system. See Connecting to the Target System topic for description of each connection method.

The 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 3.  CW TAP probe - electrical characteristics**

| Electrical Characteristics | |
|---|---|
| Target voltage levels supported | 1.2V to 3.3V; 5V tolerant |
| probe power consumption from target | Less than 50 mA 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 probe.

**Table 4.  CodeWarrior 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** | |
| **probe dimensions** | |
| *Table continues on the next page...* | |

**Table 4.  CodeWarrior TAP probe - physical characteristics (continued)**

| Physical Characteristics | |
|---|---|
| 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) |

# Chapter 6
# CodeWarrior TAP Probe Setup Utility Commands

This chapter explains how to access the CodeWarrior TAP probe internal setup utility.

It describes all available setup utility commands and arguments.

This chapter contains the following topics:

- Connecting to CodeWarrior TAP probe setup utility
- CodeWarrior TAP probe setup utility commands and variables

> **NOTE**
> The commands described in this chapter are for reference only. For detailed procedures on using these commands, see specific chapter that covers the topic of interest.

## 6.1  Connecting to CodeWarrior TAP probe setup utility

There are two methods for accessing the CodeWarrior TAP probe internal setup utility.

To connect to the setup utility:

- Connect to the CodeWarrior TAP probe USB port. Use this method if the CodeWarrior TAP probe is not connected to your network. For more information, see Customizing CodeWarrior TAP probe topic.

- Telnet to the CodeWarrior TAP probe through an existing Ethernet connection. Use this method if the CodeWarrior 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 CodeWarrior 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 CodeWarrior TAP probe.

    ```
    telnet hostname
    ```

    Use the hostname that you entered into the hosts database file, for more details, see Connecting to network topic.

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 CodeWarrior 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.

## 6.2  CodeWarrior TAP probe setup utility commands and variables

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

# 6.2.1 Commands to configure communications

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

## 6.2.1.1 netparam

The `netparam` command displays or sets non-volatile networking parameters stored in the flash EPROM of CodeWarrior 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, see Configuring CodeWarrior 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]
```

**Table 5.  Netparam parameters**

| Field | Description |
|---|---|
| **\<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 to use for `host`, specified in dotted-decimal notation |
| **add_route host gateway hop_#** | Adds a route to CodeWarrior 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 CodeWarrior TAP probe and destination host or network |
| *Table continues on the next page...* | |

**Table 5. Netparam parameters (continued)**

| Field | Description |
|---|---|
| `bootconfig {static | dhcp }[:host]` | Sets the IP address resolution protocol. It determines the boot method of CodeWarrior TAP probe. Use `bootconfig` to connect to the network either by DHCP or by storing the IP address in the flash EPROM of CodeWarrior 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 FSLXXYYZZ, where XXYYZZ is the last three octets of the MAC address, provided on a label on the bottom side of the probe. For example, if the probe's MAC address is `00:00:f6:00:77:31`, the default host name will be `FSL007731`. `static` - Use IP address stored in CodeWarrior 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` - Destination IP address of host or host network |
| `delete_route` | Deletes a route from the static route table `host`- Destination IP address of host or host network |
| `static_ip_address address [:mask]` | Sets the CodeWarrior TAP probe IP address and optional netmask `address` - IP address in dotted-decimal format (for example, 128.8.1.1). When entering the IP address by itself (without also entering the netmask), the CodeWarrior TAP probe uses the standard netmask assigned to that IP address. `mask` - Netmask in dotted-decimal format (for example, 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 (for example, 128.1.1). The DNS server at this address will be used for domain name resolution when bootconfig is set to static. |

## 6.2.1.2  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
```

**Table 6. Default target settings**

| 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 |

## 6.2.2  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 topic.

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

## 6.2.2.1  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

**Table 7. ARP - options**

| Field | Description |
|---|---|
| `-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 |

## 6.2.2.2  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 CodeWarrior TAP probe. Without options, it displays the current host table.

Syntax

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

Options

**Table 8.  Host - options**

| Field | Description |
|---|---|
| `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. |

## 6.2.2.3  netstat

Displays network information and statistics.

Syntax

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

Options

**Table 9.  Netsat - options**

| Field | Description |
|---|---|
| `-a --inet` | Display network connections |
| `-i` | Display device status |
| `-s` | Display protocol statistics |
| `-r` | Display route table |

## 6.2.2.4  ping

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

Syntax

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

**Table 10.  Ping - options**

| Field | Description |
|-------|-------------|
| hostname | Use the `hostname` stored in CodeWarrior 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. |

# 6.2.2.5  route

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

Syntax

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

Options

**Table 11.  Route - options**

| Field | Description |
|-------|-------------|
| add destination gateway | Add a dynamic route to the route table. |
| delete destination | Delete a dynamic route from the route table. |

# Chapter 7
# Network Administration

This chapter guides the network administrators in installing a CodeWarrior TAP probe.

The CodeWarrior TAP probe is an 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 chapter contains the following sections:

- CodeWarrior TAP probe network ports
- Configuring CodeWarrior TAP probe using netparam
- Using CCS to search for CodeWarrior TAP probes

## 7.1  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 12.  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 |

## 7.2  Configuring CodeWarrior 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 CodeWarrior TAP probe. Each time you enter a `netparam` command, wait for the `core>` prompt to re-appear before entering the next command.

## 7.2.1  Configuring dynamic IP address

To configure a dynamic IP address:

1. Connect to the CodeWarrior TAP probe internal setup utility, as explained in the Connecting to CodeWarrior TAP probe setup utility topic.

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.

## 7.2.2  Configuring static IP address

If you do not have a DHCP server on your network or you prefer to manually configure your network settings, the CodeWarrior TAP probe is capable of storing its IP address and netmask in flash memory. When bootconfig is set to static, the CodeWarrior 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 CodeWarrior 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 `host` file:



**Figure 15.    Host file**

---
**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 CodeWarrior 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.

## 7.2.3 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

### 7.2.3.1 Specify 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` .

## 7.2.4 Changing existing route entry

**NOTE**

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

Before entering the static routes, make a map of all gateway paths between the CodeWarrior 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.

## 7.2.5 Entering static routes

**NOTE**

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

**CodeWarrior TAP Probe User Guide, Rev. 10.x, 02/2017**

Before entering the static routes, make a map of all gateway paths between the CodeWarrior 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 CodeWarrior TAP probe crosses when communicating with the destination host. The `hop_#` is the decimal number of gateways between the CodeWarrior TAP probe and the destination host. For more details on `netparam` command, see CodeWarrior TAP probe setup utility commands topic.

2. Add routes until all destination hosts or networks are defined.

3. When the `core>` prompt returns, reset the CodeWarrior TAP probe by cycling power, or by entering the `reset` command.

# 7.2.5.1  Static route example

Three Class C networks connected by a single router shows three class "C" networks joined together by a single IP router, making each CodeWarrior TAP probe accessible from three workstations (elmer, tweety, and brutus).

No static routing information is required to make a CodeWarrior TAP probe accessible from a workstation local to it on a network. For example, the CodeWarrior 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 CodeWarrior TAP probe for each workstation on a non-local network that accesses it. The CodeWarrior 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 CodeWarrior 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
```
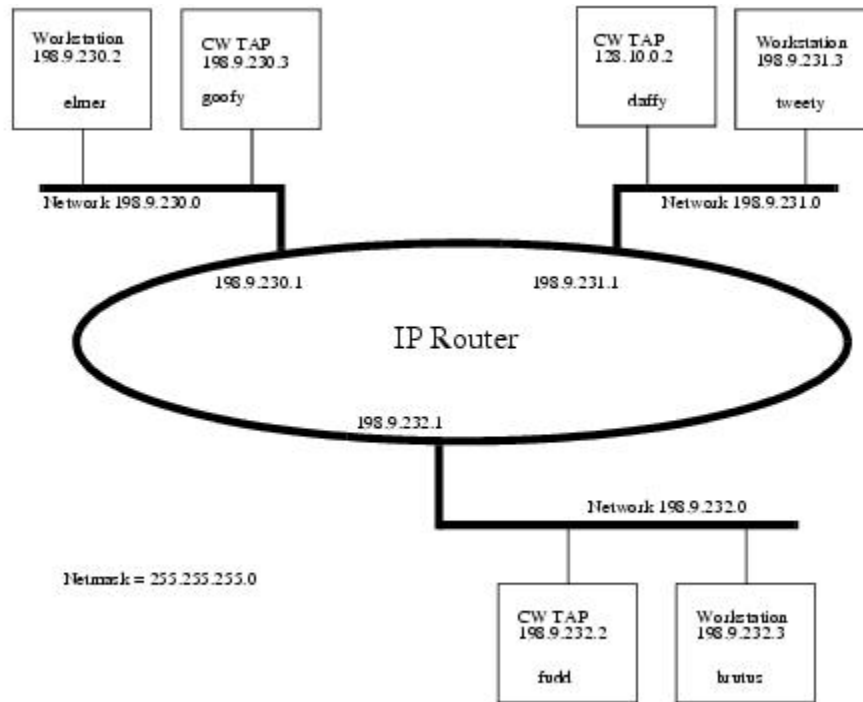
**Figure 16. Three Class C networks connected by a single router**

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

# 7.3 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 CCS icon topic) 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 17.        CCS icon**

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

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

## 7.3.1  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

%
```

# Chapter 8
# CodeWarrior TAP Probe Firmware (Core)

This chapter explains the methods for reprogramming the Boot Loader and Operating System images stored in the flash EPROM of the CodeWarrior TAP probe.

Before reprogramming the flash EPROM, make sure you have already configured the CodeWarrior TAP probe network communication.

This chapter contains the following sections:

* Boot Loader

* Reprogramming CodeWarrior TAP probe firmware images

## 8.1  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, see Reprogramming firmware through Ethernet port topic.

### 8.1.1  Operating system

The CodeWarrior 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 CodeWarrior TAP probe finishes executing the Boot Loader, it loads the OS. This is indicated by the `core` prompt in the CodeWarrior TAP probe's setup utility, and by flashing a orange or green heartbeat LED. To reprogram the OS image stored in the CodeWarrior TAP probe flash EPROM, see Reprogramming CodeWarrior TAP probe firmware images topic.

### 8.1.2  Shell software

The CodeWarrior 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.

## 8.2  Reprogramming CodeWarrior TAP probe firmware images

You need to reprogram CodeWarrior TAP probe firmware images when you are installing an update to existing software

At some point, you may be required to reprogram the CodeWarrior 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 CodeWarrior TAP probe firmware images stored in its flash EPROM.

**CodeWarrior TAP Probe User Guide, Rev. 10.x, 02/2017**

NXP Semiconductors · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · · **43**

## 8.2.1  Reprogramming Firmware through Ethernet Port

In order to use the following instructions, the CodeWarrior TAP probe communications must already be configured (see Connecting to network topic).

To reprogram the firmware image:

1.  Launch CCS and open the CCS command window. For information on launching CCS, see 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 CodeWarrior 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 CodeWarrior TAP probe that you plan to use with the current version of debugger.

# Chapter 9
# 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 18. CodeWarrior TAP probe for JTAG/COP connector pin assignments on page 45 shows the pin assignments of the probe JTAG/COP connector.

Table 13. CodeWarrior TAP probe for JTAG/COP signal directions on page 45 lists JTAG/COP signal names, direction, pin numbers, descriptions, and drive capabilities for the probe JTAG/COP connector.

Table 14. CodeWarrior TAP probe for JTAG/COP signal recommendations and requirements on page 46 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 CodeWarrior TAP probe and the target system, the JTAG/COP signals must meet the requirements listed in Table 14. CodeWarrior TAP probe for JTAG/COP signal recommendations and requirements on page 46.
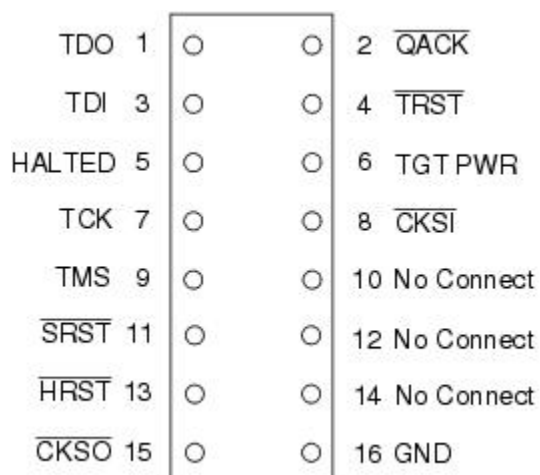


**Figure 18.  CodeWarrior TAP probe for JTAG/COP connector pin assignments**

**Table 13.  CodeWarrior TAP probe for JTAG/COP signal directions**

| JTAG/COP pin | Signal mnemonic | Signal direction | Description |
|---|---|---|---|
| 1 | TDO | From target system | 17pF load |
| 2 | QACK_B | From CodeWarrior TAP probe connector | 100ohm pull-down |
| 3 | TDI | From CodeWarrior TAP probe connector | 50mA driver |
| 4 | TRST_B | From CodeWarrior TAP probe connector | 50mA driver |
| 5 | HALTED | From target system | 17pF load |
| *Table continues on the next page...* | | | |

**Table 13.  CodeWarrior TAP probe for JTAG/COP signal directions (continued)**

| JTAG/COP pin | Signal mnemonic | Signal direction | Description |
|---|---|---|---|
| 6 | TGT PWR | From target system | 2Mohm pull-down, plus 0.01F load |
| 7 | TCK | From CodeWarrior TAP probe connector | 50mA driver |
| 8 | CKSI | From CodeWarrior TAP probe connector | 50mA driver |
| 9 | TMS | From CodeWarrior TAP probe connector | 50mA driver |
| 10 | No Connect | - n/a - | |
| 11 | SRST_B | Bi-directional | Open-drain. 5ohm to ground when asserted by CodeWarrior TAP probe, 22pF load when not asserted |
| 12 | No Connect | - n/a - | |
| 13 | HRST_B | Bi-directional | Open-drain. 5Ohm to ground when asserted by CodeWarrior TAP probe, 22pF load when not asserted |
| 14 | No Connect | - n/a - | |
| 15 | CKSO_B | From target system | 17pF load |
| 16 | GND | - n/a - | |

.

**Table 14.  CodeWarrior 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 CodeWarrior 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. |
| | | *Table continues on the next page...* |

---

 [1]  14.7KOhm pull-up to buffered TGT PWR.

**Table 14.  CodeWarrior TAP probe for JTAG/COP signal recommendations and requirements (continued)**

| JTAG/COP pin | Signal mnemonic | Requirement |
|---|---|---|
| 2 | QACK_B | May be wired to the target system processor. QACK_B is an input to most PowerPC processors and must remain low while the CodeWarrior TAP probe is connected to the target system. The CodeWarrior 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 system processor. The CodeWarrior 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. |
| 4 | TRST_B | Must be wired to the target system processor. The CodeWarrior TAP probe drives the TRST_B output with up to 50mA. To gain control of the processor, the CodeWarrior TAP probe negates TRST_B approximately 250ms before negation of HRST_B. This allows the CodeWarrior TAP probe to issue COP commands through the JTAG/COP interface and gain control of the processor upon negation of HRST_B. The TRST_B 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 CodeWarrior TAP probe does not currently use this signal. |

*Table continues on the next page...*

**Table 14.  CodeWarrior TAP probe for JTAG/COP signal recommendations and requirements (continued)**

| JTAG/COP pin | Signal mnemonic | Requirement |
|---|---|---|
| 6 | TGT PWR | Must be wired to the target system. The CodeWarrior 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 CodeWarrior TAP probe (CKSI_B, TRST_B, 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 50µA from this signal, so a weak pull-up is sufficient (1KOhm). |
| 7 | TCK | Must be wired to the target system processor. The CodeWarrior 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 CodeWarrior TAP probe does not currently use this signal. |
| 9 | TMS | Must be wired to the target system processor. The CodeWarrior 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 |
| 11 | SRST_B | May be wired to the target system processor. During reset, the CodeWarrior TAP probe drives SRST_B to ground through a 5Ohm resistor. |
| 12 | No Connect | Not required for emulation |
| 13 | HRST_B | Must be wired to the target system processor. During reset, the CodeWarrior TAP probe drives HRST_B to ground through a 5Ohm resistor. |

*Table continues on the next page...*

**Table 14.  CodeWarrior TAP probe for JTAG/COP signal recommendations and requirements (continued)**

| JTAG/COP pin | Signal mnemonic | Requirement |
|---|---|---|
| 14 | No Connect | Not required for emulation |
| 15 | CKSO_B | Should be wired to the target system processor. The CodeWarrior TAP probe senses CKSO_B 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 CodeWarrior TAP probe. |

# Chapter 10
# 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 19. CodeWarrior TAP probe OnCE connector pin assignments on page 50 shows the pin assignments of the probe OnCE connector.

Table 15. CodeWarrior TAP probe OnCE signal directions on page 50 lists OnCE signal names, direction, pin numbers, descriptions, and drive capabilities for the probe OnCE connector.

Table 16. CodeWarrior TAP probe OnCE signal recommendations and requirements on page 51 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 CodeWarrior TAP probe and the target system, the OnCE signals must meet the requirements listed in Table 16. CodeWarrior TAP probe OnCE signal recommendations and requirements on page 51.
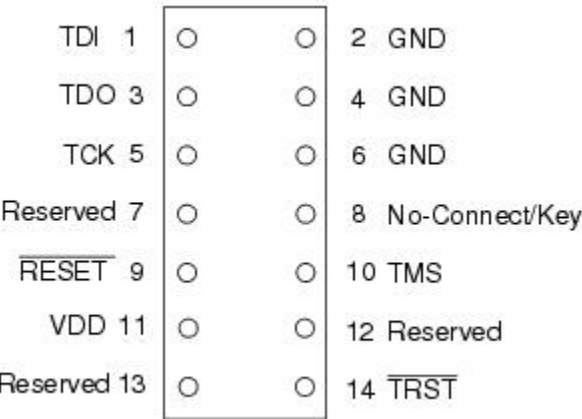


**Figure 19.  CodeWarrior TAP probe OnCE connector pin assignments**

**Table 15.  CodeWarrior TAP probe OnCE signal directions**

| OnCE pin | Signal mnemonic | Signal direction | Description |
|---|---|---|---|
| 1 | TDI | From CodeWarrior TAP probe connector | 50mA driver |
| 2 | GND | - n/a - | |
| 3 | TDO | From target system | 17pF load |
| 4 | GND | - n/a - | |
| 5 | TCK | From CodeWarrior TAP probe connector | 50mA driver [2] |

*Table continues on the next page...*

---

[2]  14.7Kohm pull-up to buffered VDD.

**Table 15. CodeWarrior TAP probe OnCE signal directions (continued)**

| OnCE pin | Signal mnemonic | Signal direction | Description |
|---|---|---|---|
| 6 | GND | - n/a - | |
| 7 | Reserved | From CodeWarrior TAP probe connector | 50mA driver |
| 8 | No-Connect/Key | - n/a - | |
| 9 | RESET_B | Bi-directional | Open-drain. 5ohm to ground when asserted by CodeWarrior TAP probe, 22pF load when not asserted[2] |
| 10 | TMS | From CodeWarrior TAP probe connector | 50mA driver |
| 11 | VDD | From target system | 2Mohm pull-down, plus 0.01uF load |
| 12 | Reserved | Bi-directional | Open-drain, 5ohm to ground when asserted by CodeWarrior TAP probe, 22pF load when not asserted[2] |
| 13 | Reserved | From target system | 17pF load |
| 14 | TRST_B | From CodeWarrior TAP probe connector | 50mA driver |

**Table 16. CodeWarrior TAP probe OnCE signal recommendations and requirements**

| OnCE pin | Signal mnemonic | Requirement |
|---|---|---|
| 1 | TDI | Must be wired to the target system processor. The CodeWarrior 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 CodeWarrior TAP probe. |

*Table continues on the next page...*

**Table 16. CodeWarrior TAP probe OnCE signal recommendations and requirements (continued)**

| OnCE pin | Signal mnemonic | Requirement |
|---|---|---|
| 3 | TDO | Must be wired to the target system processor. TDO is an output from the target system processor and input to the CodeWarrior 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 CodeWarrior TAP probe. |
| 5 | TCK | Must be wired to the target system processor. The CodeWarrior 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 CodeWarrior TAP probe. |
| 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 | RESET_B | Must be wired to the target system processor. During reset, the CodeWarrior TAP probe drives RESET_B to ground through a 5ohm resistor. |
| 10 | TMS | Must be wired to the target system processor. The CodeWarrior 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. |

*Table continues on the next page...*

**Table 16.  CodeWarrior TAP probe OnCE signal recommendations and requirements (continued)**

| OnCE pin | Signal mnemonic | Requirement |
|---|---|---|
| 11 | VDD | Must be wired to the target system. The CodeWarrior 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 CodeWarrior TAP probe (TDI, TCK, TMS, RESET_B, and TRST_B). |
| 12 | Reserved | Not required for emulation. |
| 13 | Reserved | Not required for emulation. |
| 14 | TRST_B | Must be wired to the target system processor. The CodeWarrior TAP probe drives the TRST_B output with up to 50 mA. The TRST_B trace run should be kept short, and should maintain a "two-signal-width" spacing from any other parallel dynamic signal trace. |

—

# Chapter 11
# ARM CoreSight Connector Information

Figure 20. CodeWarrior TAP probe for ARM connector pin assignments on page 54 image shows the pin assignments of the probe ARM CoreSight connector.

Table 17. CodeWarrior TAP probe for ARM signal directions on page 54 lists ARM signal names, direction, pin numbers, descriptions, and drive capabilities for the probe ARM CoreSight connector.

Table 18. CodeWarrior TAP probe for ARM signal recommendations and requirements on page 55 provides a general description of each ARM signal and the operational requirements.

> **NOTE**
>
> All ARM signals must meet accepted standards for ARM signal design. To ensure proper and stable operation between the CodeWarrior TAP probe and the target system, the ARM signals must meet the requirements listed in Table 18. CodeWarrior TAP probe for ARM signal recommendations and requirements on page 55
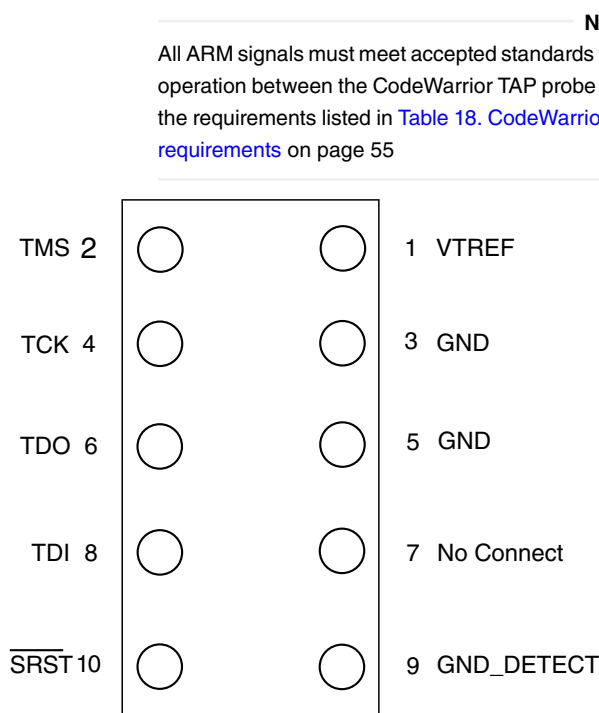


**Figure 20. CodeWarrior TAP probe for ARM connector pin assignments**

**Table 17. CodeWarrior TAP probe for ARM signal directions**

| ARM pins | Signal mnemonics | Signal directions | Description |
|---|---|---|---|
| 1 | VTREF | From target system | 2Mohm pull-down, plus 0.01F load |
| 2 | TMS | From CodeWarrior TAP probe connector | 50mA driver |
| 3 | GND | - n/a - | - |
| 4 | TCK | From CodeWarrior TAP probe connector | 50mA driver |
| 5 | GND | - n/a - | - |
| 6 | TDO | From target system | 17pF load |
| *Table continues on the next page...* | | | |

**Table 17. CodeWarrior TAP probe for ARM signal directions (continued)**

| ARM pins | Signal mnemonics | Signal directions | Description |
|----------|------------------|-------------------|-------------|
| 7 | No Connect | - n/a - | - |
| 8 | TDI | From CodeWarrior TAP probe connector | 50mA driver |
| 9 | GND_DETECT | From CodeWarrior TAP probe connector | Grounded by CodeWarrior TAP |
| 10 | SRST_B | Bi-directional | Open-drain. 5ohm to ground when asserted by CodeWarrior TAP probe, 22pF load when not asserted |

**Table 18. CodeWarrior TAP probe for ARM signal recommendations and requirements**

| ARM pins | Signal mnemonics | Requirements |
|----------|------------------|--------------|
| 1 | VTREF | Must be wired to the target system. The CodeWarrior 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 CodeWarrior TAP probe (CKSI_B, TRST_B, 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 50µA from this signal, so a weak pullup is sufficient (1KOhm). |
| 2 | TMS | Must be wired to the target system processor. The CodeWarrior TAP probe drives the TMS output with up to 50mA. TMS should be kept as short as possible and maintain a "twosignal-width" spacing from any other parallel dynamic signal trace. TMS should have a termination option at the processor. |
| 3 | GND | Must be wired to the target system. GND is connected directly to the ground inside the CodeWarrior TAP probe. |
| 4 | TCK | Must be wired to the target system processor. The CodeWarrior 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. |

*Table continues on the next page...*

**Table 18.  CodeWarrior TAP probe for ARM signal recommendations and requirements (continued)**

| ARM pins | Signal mnemonics | Requirements |
|---|---|---|
| 5 | GND | Must be wired to the target system. GND is connected directly to the ground inside the CodeWarrior TAP probe. |
| 6 | TDO | Must be wired to the target system processor. TDO is an output from the target system processor and an input to the CodeWarrior TAP probe. The TDO trace run should be kept short, and should maintain a "two-signalwidth" spacing from any other parallel dynamic signal trace. TDO should have a series termination resistor located near the target system processor. |
| 7 | No Connect | Not required for emulation. |
| 8 | TDI | Must be wired to the target system processor. The CodeWarrior TAP probe drives the TDI output with up to 50mA. The TDI trace should be kept short, and should maintain a "twosignal- width" spacing from any other parallel dynamic signal trace. TDI should have an RC termination option at the processor. |
| 9 | GND_DETECT | Target can use a pull-up to detect presence of a JTAG probe. If this functionality is not needed, targets should also ground this pin. |
| 10 | SRST_B | May be wired to the target system processor. During reset, the CodeWarrior TAP probe drives SRST_B to ground through a 5Ohm resistor. |

# Chapter 12
# Troubleshooting

This chapter provides CodeWarrior TAP probe troubleshooting information.

This chapter contains the following topics:

- Troubleshooting communications problems
- Troubleshooting power problems

## 12.1 Troubleshooting communications problems

You will understand how to troubleshoot the communication problems between the debugger and the CodeWarrior TAP probe.

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

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

  The CodeWarrior 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 CodeWarrior TAP probe is receiving power. See LEDs on CodeWarrior TAP probe topic for a description of the status LEDs.

- Make sure the CodeWarrior TAP probe is running the OS software. For more information on loading the OS software, see CodeWarrior TAP probe firmware (Core) topic.

- Use the communication troubleshooting utilities of CodeWarrior TAP probe to verify that it is recognized on your network, or to help diagnose problems connecting to your network.

  To troubleshoot communication, see Verify network communication topic. To list all the CodeWarrior probes on your local subnets, use the CCS `findcc` host utility. See Using CCS to search for CodeWarrior TAP probes topic.

- Make sure the debugger is set up correctly for Ethernet communication with the CodeWarrior TAP probe.

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

## 12.1.1 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 CodeWarrior TAP probe.

To verify network communication:

1. Connect to the CodeWarrior TAP probe internal setup utility, see Connecting to CodeWarrior TAP probe setup utility topic.

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 CodeWarrior TAP probe may not automatically recognize the hostname. To ping a hostname, the CodeWarrior TAP probe internal host table must first be updated.

---

## 12.1.2  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 CodeWarrior 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 netstat topic.

## 12.2  Troubleshooting power problems

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

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

# Index

CWTAPUG
Rev. 10.x
02/2017

# OCEAN CHIPS
## Океан Электроники
### Поставка электронных компонентов

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics  и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «**JONHON**», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «**FORSTAR**».



**«JONHON»** (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

**«FORSTAR»** (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели,
кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).



Телефон:     8 (812) 309-75-97 (многоканальный)
Факс:          8 (812) 320-03-32
Электронная почта:   ocean@oceanchips.ru
Web:     http://oceanchips.ru/
Адрес:   198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А