

ARM

series

SIGNUM SYSTEMS CORPORATION

ARM RDI 1.5.1 Driver for OMAP

Installation Instructions

SIGNUM
S Y S T E M S

COPYRIGHT NOTICE

Copyright (c) 2003 by Signum Systems Corporation. All rights are reserved worldwide. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of Signum Systems.

DISCLAIMER

Signum Systems makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Also, Signum Systems reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Signum Systems to notify any person or organization of such revision or changes.

WARRANTY

Signum Systems warrants to the original purchaser that this product is free of defects in material and workmanship and performs to applicable published Signum Systems specifications for a period of SIX MONTHS from the date of shipment. If defective, the product must be returned to Signum Systems, prepaid, within the warranty period, and it will be repaired or replaced (at our option) at no charge. Equipment or parts which have been subject to misuse, abuse, alteration, neglect, accident, unauthorized installation or repair are not covered by warranty. This warranty is in lieu of any other warranty expressed or implied. **IN NO EVENT SHALL SIGNUM SYSTEMS BE LIABLE FOR CONSEQUENTIAL DAMAGES OF ANY KIND.** It is up to the purchaser to determine the reliability and suitability of this product for his particular application.

SIGNUM
S Y S T E M S
11992 CHALLENGER COURT
MOORPARK, CA 93021, U.S.A
PHONE 805 • 523 • 9774
WWW.SIGNUM.COM

Purpose *This document describes the installation process for the RDI (Remote Debug Interface) 1.5.1 driver used with the Signum emulator for the OMAP processor.*

Installing the Driver

1. Insert the *Development Tools for Microsoft Windows* CD into the CD-ROM drive. In the Master Setup dialog box, select RDI 1.5.1 Driver for Signum emulators.

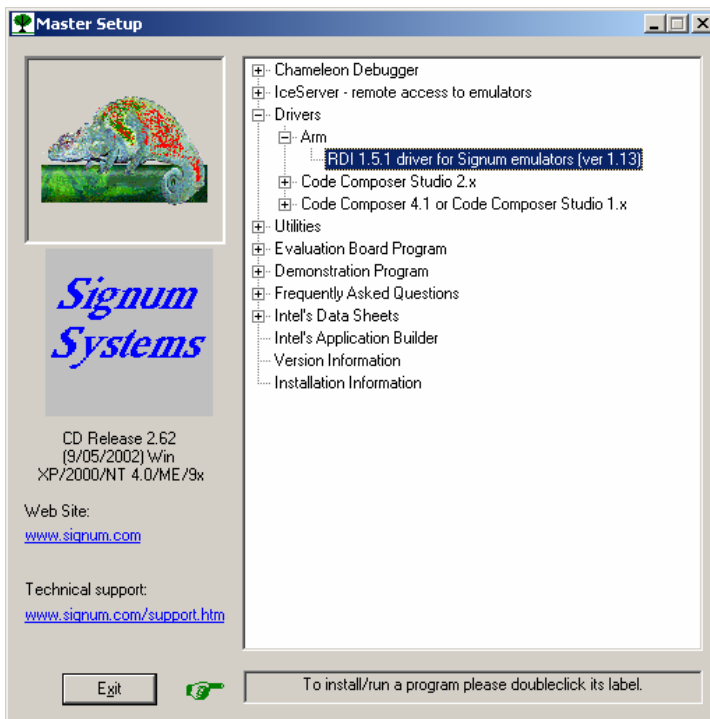


FIGURE 1 Master Setup dialog box.

S I G N U M S Y S T E M S

2. When the Setup program starts, select the program folder for the program icons and the destination folder for the driver to be installed in.

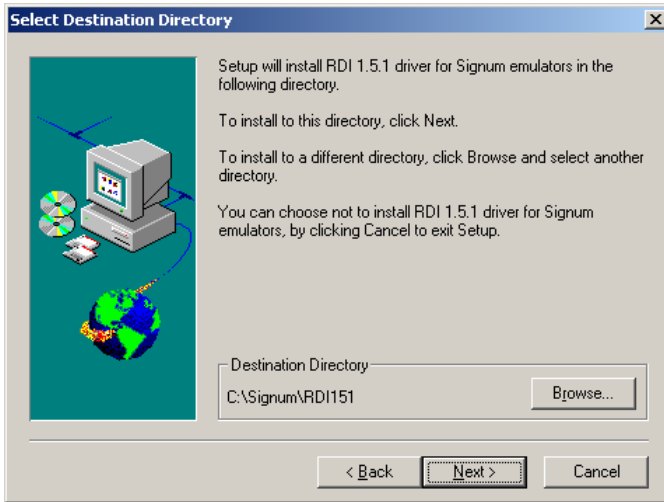


FIGURE 2 Selecting the folder for the driver.

In the confirmation dialog box, press the Next button to copy the driver files to your hard drive (Figure 3).

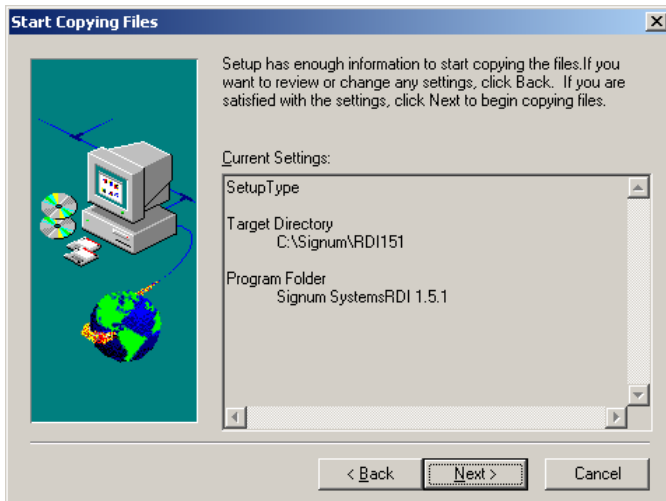


FIGURE 3 Confirming the driver location.

3. If you are running Windows XP, 2000 or NT 4.0, the parallel port driver (ParalUSP.sys), necessary to communicate with the emulator is installed. Restart your system if the Setup program asks you to do so.
4. Complete the installation process including the system re-start on Windows XP, 2000 and NT systems. Connect the emulator to a parallel port of your computer. For best performance, make sure that the port is configured for the EPP mode — the standard mode is less efficient. Currently, the emulator does not support the ECP mode.
5. Connect the JTAG cable to the JTAG connector on your OMAP system or evaluation board. Turn the emulator on first, then turn on your target board. Finally, configure your debugger, as described in the next section.

Configuring the Debugger

The ARM RDI 1.5.1 driver for Signum emulators is compatible with all the ARM debuggers that conform to the RDI 1.5.1 specification, including among others:

- ARM extended Debugger (AXD) for Windows,
- ARM Debugger for Windows (ADW) SDT 2.51 and higher,
- ARM Debugger for Windows (ADW) ADS v1.0.1 and higher.

The following debugger configuration procedure uses the AXD debugger as an example. In most situations, extending this procedure to other debuggers will be straightforward.

1. Run the AXD debugger.
2. From the debugger's Options menu, select Configure Target. The Choose Target dialog appears (Figure 4).

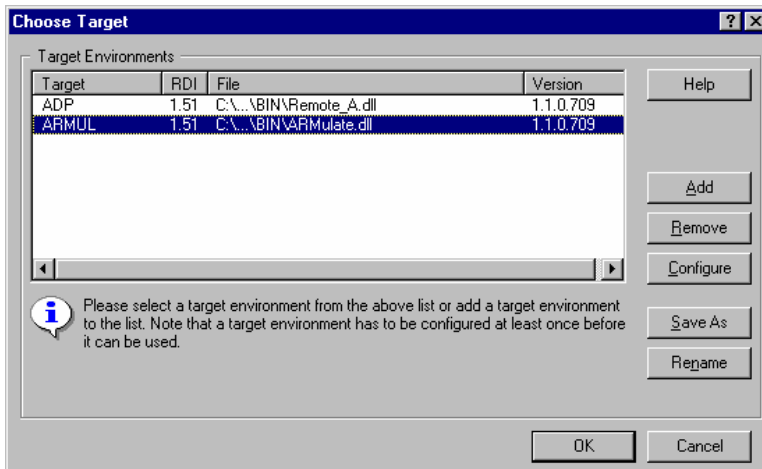


FIGURE 4 Selecting the driver DLL.

Press the Add button. In the Windows Open dialog, navigate to the ARM RDI 1.5.1 driver file. If you accepted the defaults from the *Installing the Driver* section, the path to the driver file should be C:\Signum\RDI151\SigJdsRDI.dll. Click the Open button. The driver is now listed in the Choose Target dialog box (Figure 5).

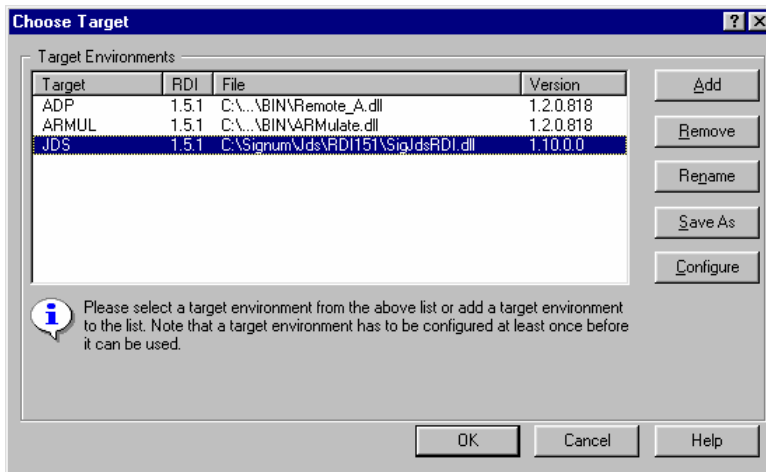


FIGURE 5 The RDI driver for the Signum emulator is now available.

3. Press the Configure button to configure the connection between your debugger and the Signum emulator.

Configuring the Connection

1. In the JDS Configuration dialog that appears, make sure that the Connection tab is selected (Figure 6).

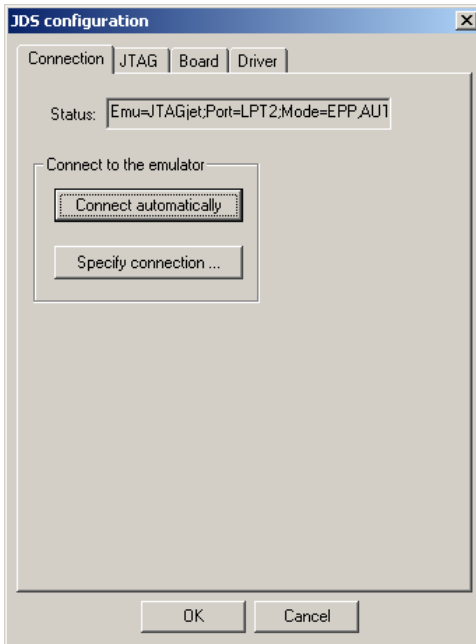


FIGURE 6 Configuring the emulator: the Connection tab.

- To establish a debugger-emulator connection with minimum effort, press the Connect Automatically button. The connection parameters appear in the Status box.
 - For added control over the connection, press the Specify Connection button to set the communication port and its parameters manually.
2. Once the connection has been established, in the JTAG configuration dialog, select the JTAG tab. Choose Specify JTAG Configuration File (Figure 7).

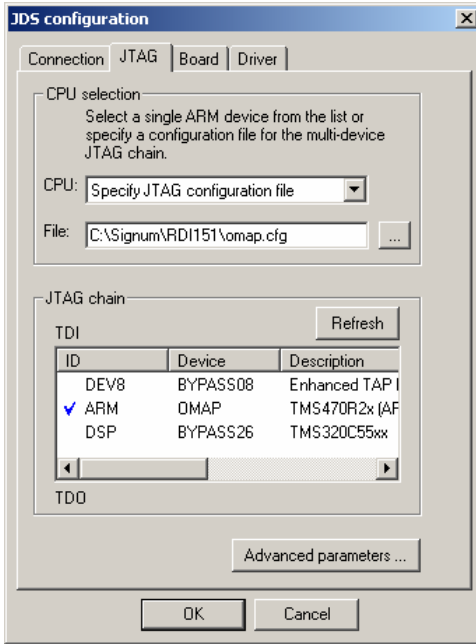


FIGURE 7 Configuring the emulator: the JTAG tab.

Click the file browser button located next to the File text box and navigate to the file omap.cfg. After the file has been loaded, the JTAG chain group box displays the chain description. In the ID column, select ARM as the device to be debugged. Note that you cannot select a bypass device.

3. Click the Advanced Parameters button and select the TI mode from the Emulation Mode drop-down list (see Figure 8).

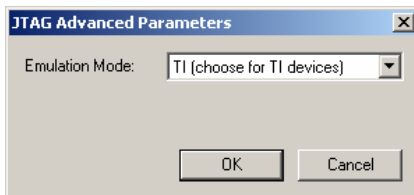


FIGURE 8 Selecting the emulation mode for OMAP.

4. Click the Board tab and browse for the startup macro file Innovator.mac or OMAP1510EVM.mac, depending on the target board you are using (Figure 9).

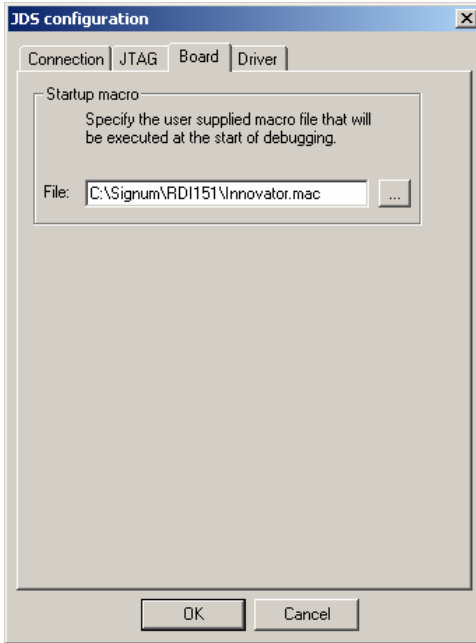


FIGURE 9 Configuring the emulator: the Board tab.

5. Click the Driver tab to configure driver protocol logs and error reports.
 - To display the driver protocol log in the debugger Log window, select Log Enable. If you also need to store the log in a file, enter the file name in the File text box.
 - To enable the driver to generate descriptive error messages, select Show Error Messages (Figure 10). This option does not affect the way your debugger displays its own error messages; it is designed simply to augment and clarify those debugger messages that tend to be cryptic or are limited to error codes only.

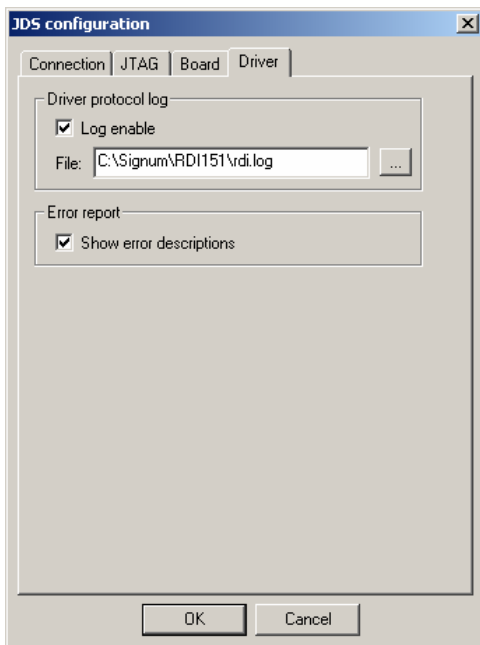


FIGURE 10 Configuring the emulator: the Driver tab.

6. Click OK. In the Choose Target dialog box, click OK again to accept your settings and connect to the target processor.

JTAG Chain Configuration File

A multi-device JTAG daisy chain is configured using the so-called JTAG chain configuration file. This file is an ASCII text file with the file name extension `.cfg`. With the exception of comment lines, each line in a `.cfg` file refers to a separate device. Thus in general, the file format is as follows:

```

ID1   Device1   Description1
ID2   Device2   Description2

```

S I G N U M S Y S T E M S

.
. .
etc.,

where

<i>ID</i>	Is a unique name identifying the device, e.g., "CPU_1", including the double quotes.
<i>Device</i>	Is the type of the device, such as ARM7TDMI. Non-ARM devices should be bypassed, and specified as BYPASSxx, where xx denotes the length of the instruction register in a two-digit format. For instance, BYPASS0A denotes a bypassed device with a 10-bit (0a hex) instruction register.
<i>Description</i>	Is a coment text.

In general, enclose in double quotes all names, words or phrases that contain non-alphanumeric characters. Lines that begin with the semicolon (;) are treated as comments and are ignored. The order in which the JTAG devices are specified in the configuration file is significant: the first line corresponds to the device closest to the TDI, the second one to the next device in the chain, and so on. Finally, the last line describes the device on the TDO side of the chain.

An example of the JTAG configuration file is shown below.

```
; File name:  "omap.cfg"  
; JTAG configuration file for TI OMAP multi-processor  
core  
  
"DEV8"  BYPASS08      "Enhanced TAP Linking Module"  
"ARM"   OMAP         "TMS470R2x (ARM925T)"  
"DSP"   BYPASS26     "TMS320C55xx" ; C55xx IR=38 dec  
  
; End of file "omap.cfg"
```

Board Startup Macro File

Some application boards may require to be properly set up before a debug session can begin. Use a startup macro to configure on-board memory before the debugger attempts to access that memory. The macro is also a good place for disabling the watchdog that otherwise may reset the CPU soon after the debugger starts running.

A board startup configuration file is an ASCII text file with the extension .mac. The following commands can be placed in that file (Table 1).

sd <address> = <value>	Write a <value> of DWORD (32bit) type to memory at address <address>.
sw <address> = <value>	Write a <value> of WORD (16bi) type to memory at address <address>.
sb <address> = <value>	Write a <value> of BYTE (8bit) type to memory at address <address>.
Reset	Reset the CPU.
emu <parameter> = <value>	Set one of the following emulation parameters: <ul style="list-style-type: none"> • cmdline (argument string for the debuggee) • semihosting_enabled (0 – disabled, 1 – enabled)

TABLE 1 Board startup configuration commands.

Empty lines, white spaces, and comments starting with a semicolon are ignored.

The following is an example of a typical startup macro file.

```
; Innovator.mac           - RDI Startup macro
```

SIGNUM SYSTEMS

```
;                                     for OMAP1510DC Innovator.
; Version 1.00 9/25/02 - Initial version

; Configure OMAP1510 Innovator evaluation target

; Disable ARM watchdog
sw 0xFFFFEC808 = 0x00F5
sw 0xFFFFEC808 = 0x00A0

; Place device in Sync-Scalable Mode at 120MHz
sw 0xFFFFECE18 = 0x1000           ; ARM_SYSST: Sync-Scaleable Mode
sw 0xFFFFECF00 = 0x2510           ; DPLL1_CONTROL_REG: Spin up to 120MHz
sw 0xFFFFECE00 = 0x3509           ; ARM_CKCTL: SPMMU/2, TC/2, ARM/1,
                                   DSP/1, LCD/4, PER/2

; Start ARM clocks, remove reset from ARM peripherals
sw 0xFFFFECE08 = 0x06FF           ; ARM_IDLECT2
sw 0xFFFFECE14 = 0x0001           ; ARM_RSTCT2: Remove Reset from ARM
peripherals

; Initialize memory
sd 0xFFFFECC10 = 0x00203339       ; EMIFS nCS0 configuration
sd 0xFFFFECC14 = 0x00001139       ; EMIFS nCS1 configuration
sd 0xFFFFECC18 = 0x00001139       ; EMIFS nCS2 configuration
sd 0xFFFFECC1C = 0x00001139       ; EMIFS nCS3 configuration
sd 0xFFFFECC20 = 0x00010074       ; EMIFS nCS4 configuration
sd 0xFFFFECC24 = 0x00000037       ; MRS nCS4 initialization

; Configure the emulator
emu semihosting_enabled=0         ; disable semihosting
emu StepIntoInt=0                 ; disable interrupts in single-step

; End of file 'Innovator.mac'
```

