ARM

series

SIGNUM SYSTEMS CORPORATION

Chameleon Debugger for ARM and XScale

Installation Instructions

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 SYSTEMS
1211 FLYNN RD., UNIT #104
CAMARILLO, CA 93012, U.S.A
PHONE 805 • 383 • 3682
WWW.SIGNUM.COM

Purpose *This document describes the Chameleon Debugger software installation process for use with the Signum Systems JTAGjet emulator for ARM processors. For examples of connecting the JTAGjet to selected ARM and XScale target boards, please refer to "ARM Board Setup: User Guide."*

Note:
You must install the USB driver before this step. For instructions on USB driver installation, please refer to the "USB 2.0 Driver for JTAGjet and ADM51: Installation Instructions" document.

Installation Procedure

1. Insert the *JTAGjet* CD into your CD drive. From the Master Setup window, select Chameleon Debugger, and double-click ARM and XScale, as shown in Figure 1. (This selection includes the TI OMAP processor family.)

FIGURE 1 Chameleon Master Setup window.

2. Follow the online instructions and complete the installation process.

3. Start Chameleon Debugger and click the **Add Target** button in the System Configuration window (Figure 2).

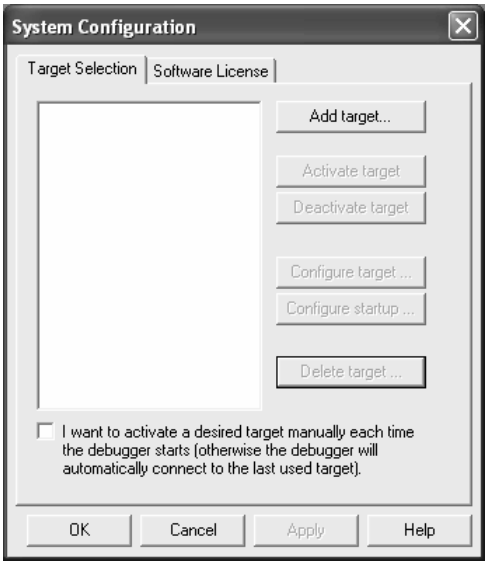

FIGURE 2 Creating a target in the debugger.

4. The Target Selection window will appear. Click the **Enter a Key** button and copy the the license key printed on a separate page titled **Product User License Certificate** to the Enter Upgrade Key box and click **OK**:

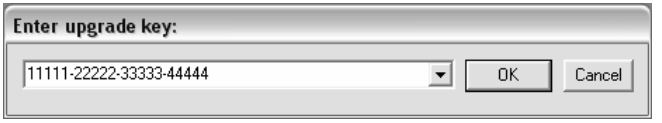

FIGURE 3 Entering the software activation/upgrade key. The actual key is supplied with the emulator package.

5. In the Target Selection dialog box, choose **ARM** as the CPU Family, and JTAGjet (Signum Systems) in the Emulator/API section (see Figure 4).

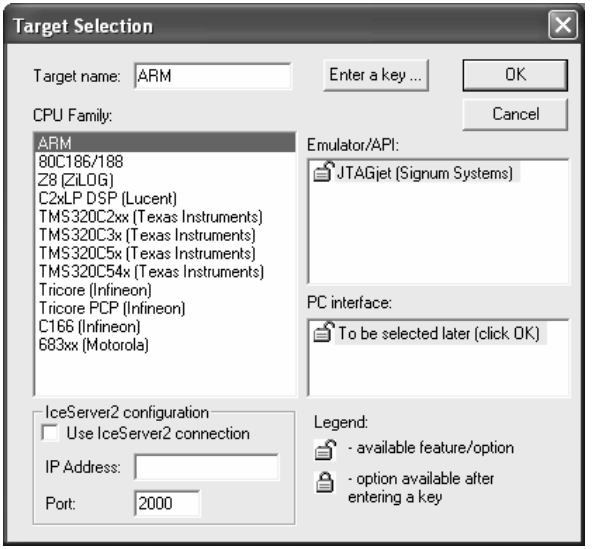

FIGURE 4 Selecting the emulator and its communication interface.

Next, please change the **Target name** (shown as ARM) to correspond with your target board. Most likely, it will be a board name or project name that will allow you to easily differentiate it from other ARM target boards. Click OK. In the dialog box that appears, select the communication port used by the JTAGjet emulator.

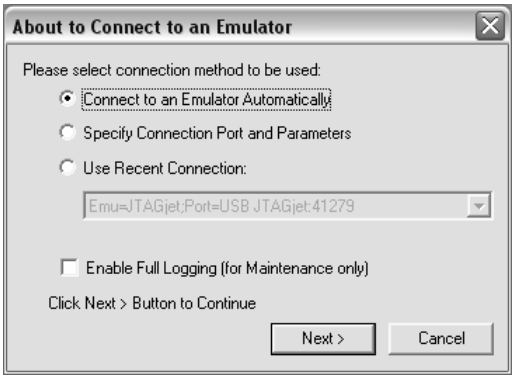

FIGURE 5 Choosing the emulator-PC communication method.

The automatic method attempts to find the fastest connection available and is hassle-free. If you need greater control over the connection, choose the "Specify Connection Port and Parameters" option and select the proper connection.

6. In the Startup Configuration Selection window a list of supported boards will appear, sorted by manufactures of the ARM device used on the board. (Figure 6).

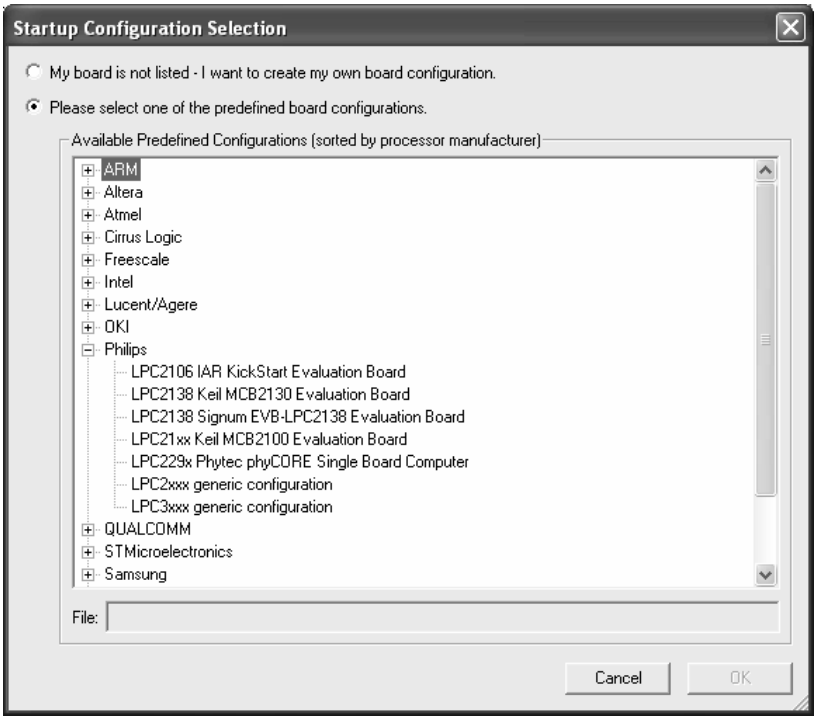

FIGURE 6 The Startup Configuration Selection window..

Please expand the device manufacturer list, select the board you have and click OK. This **completes** the Target Selection process and you will be taken to the debugger main window. If the power is applied to the board, the board will be automatically configured by a startup script (.mac macro file) and the macro file will be available as a button on the toolbar menu. Some macors have other macros inside them which may be useful during debugging. A complete list of the availabe sub-macros will be shown when you click the macro button:

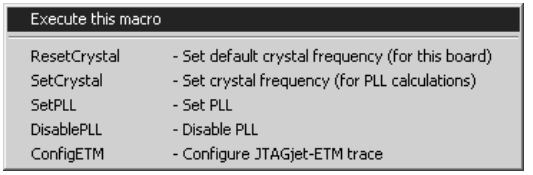

FIGURE 7

For information on other debugger features, please refer to the Help menu.

You can add other target boards to your system configuration later by clicking on the View/System Configuration menu. You may have many different boards listed in your Target Selection menu so switching between boards is easy and quick. You should **skip** the rest of this installation process.

7. Continue here if you were not able to find your board in the step above or if you have you own, unique board that requires special settings. If the baord or device you have was not listed in the window, you must select the:

◉ **My board is not listed – I want to create my own board confuration**

option at the very top of the window and click OK. A CPU/ARM core selection list will appear. It is recommended that you try to locate the name of the CPU device you use, rather then the core, as it contains more information about the device. If the CPU name is not listed, it is very important to choose the correct ARM7, ARM9 or XScale core found inside your device or ASIC. Please refer to the device data sheet to make sure you are selecting the right core.

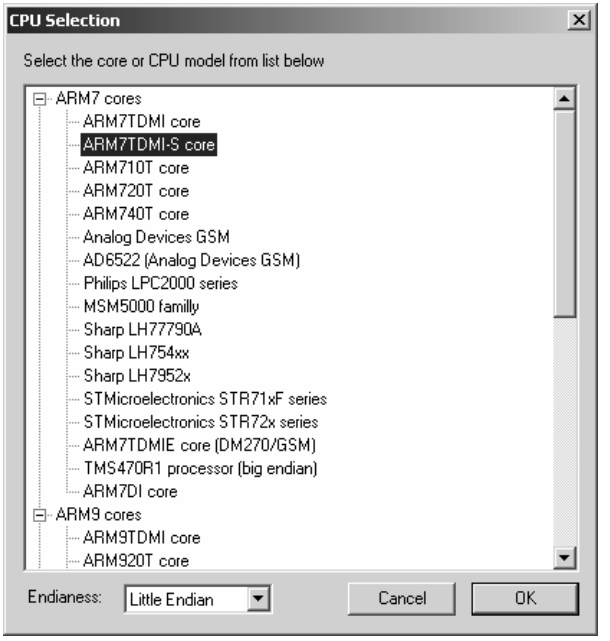

FIGURE 8 The CPU selection list.

8. When the proper core is selected, please select the **Endianess** of the board. Most ARM devices and boards are configured in the **Little Endian** mode. Again, please refer to the board manual if in doubt. Click OK to proceed.

9. The next step in target configuration needs to know JTAG Chain Geometry, i.e. how many devices are on the JTAG chain and what are their sizes.

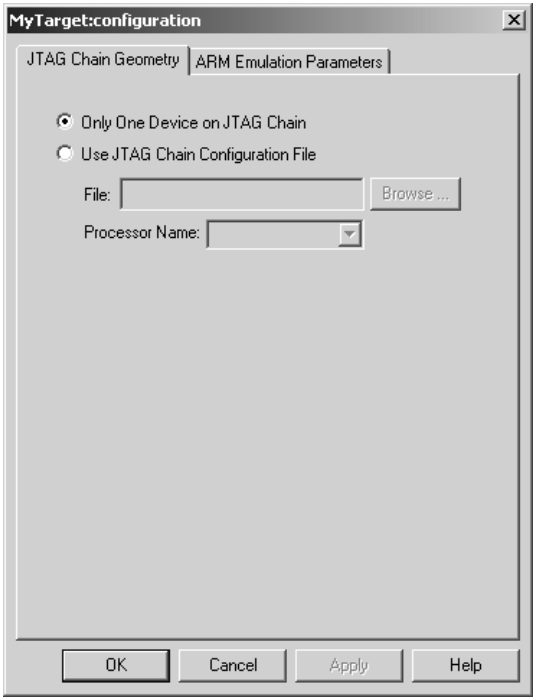

FIGURE 9 Configuring the JTAG chain.

Define the devices in the JTAG daisy chain using the following options.

◉ Only One Device on JTAG Chain Select this option if there is only one device in the chain and skip the rest of this section to continue with setup.

◉ Use JTAG Chain Configuration File If there are multiple devices in the chain, it is necessary to create an ASCII JTAG configuration file that specifies the symbolic names, types, and the length of the JTAG registers. While the file name is arbitrary, the extension must be .cfg. Each line in the .cfg file—except for comments

and empty lines—refers to a separate device on the chain. Below is the file syntax:

```
NAME1          TYPE1
NAME2          TYPE2
     .
     .
     .
```

where

`NAME` is a unique name identifying the device, e.g., "CPU_A", including the required double quotes.

`TYPE` is the type of the device, such as ARM7TDMI, ARM940T, or PXA27x. To bypass a device, specify their type as the length of the instruction register in a two-digit format appended to the word BYPASS. For example, BYPASS2e denotes a bypassed device with a 46-bit (2e hex) instruction register. If TYPE contains non-alphanumeric characters, or begins with a digit, enclose it in double quotes, as in `"ARM926EJ-S"`.

Any text between the semicolon and the end of line is treated as a comment and is ignored. The order in which the JTAG devices are specified in the configuration file is important. Namely, the first line corresponds to the device closest to the TDI, the second line corresponds to the next device in the chain, and so on. The last line refers to the device on the TDO side of the chain.

A sample JTAG configuration file is shown below.

```
;***********************************************
;* JTAG configuration file
;***********************************************

; Emulator TDI

"ARM1"    ARM7TDMI
"ARM2"    OMAP
"DSP"     BYPASS2e        ; IR=46 dec
"ARM3"    "ARM926EJ-S"

; End of file
```

When the correct JTAG chain configuration file is create, please point to this file using the **Browse** button and select the proper device in the Processor Name filed.

Filename — Specifies the JTAG configuration file. The file must reside in the directory where Chameleon Debugger was installed, normally, C:\Signum\Chameleon.

Processor name — Specifies the name of the emulated JTAG device. Choose the device that needs ot be debugged here.

10. Now click on the right tab to select the ARM Emulation Parameters. This tab allows you to decide if Semihosting (Virtual I/O) will be used and to set the JTAG clock speed, JTAG pin configuration as well as Reset timing. Note the slightly differences between the tab for ARM and XScale target boards (Figure 10).

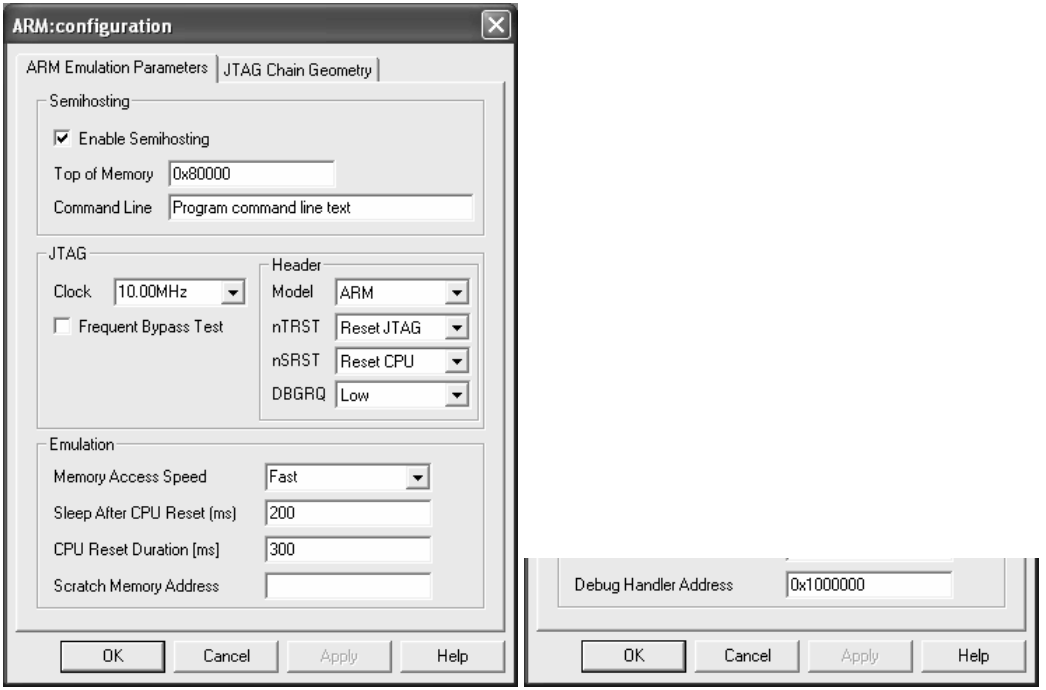

FIGURE 10 The ARM Emulation Parameters tab for ARM targets (left) and XScale targets (right).

Enable Semihosting

This enables handling of the Semihosting (Virtual I/O) by the debugger. When checked, the debugger will catch all SWI calls made by the semihosting libraries (printf, scanf, etc.) so that these calls can be redirected to the proper system calls on your Windows PC. For the Semihosting to work, you must also define the top of Memory (RAM) for the semihosting libraries that you use.

If Enable Semihosting is not checked and your application calls a semihosting library, the application will start executing code from the SWI vector location and most likely crash by executing non-existing code.

Enter the Top Of Memory parameter to reflect the size of your target board's RAM memory. This parameter allows the ARM run-time library to determine where the heap and stack should be placed in memory. For example, set top_of_memory (a hex value) to

80000 for the evaluation board ARM Evaluator 7T

– or –

9000000for the evaluation board Cogent CDK238.

If you do not use the C/C++ runtime library with semihosting support, you may wish to disable semihosting. This will make one additional hardware breakpoint on ARM7 cores available to the user.

JTAG

The JTAG section allows to select the JTAG clock frequency. If your core has the –S suffix, it internally synchonizes the JTAG TCK clock with the core clock and sends back the synchronized RTCK clock back to the emulator. This is called **Adaptive** clocking and should be used on all cores that send back the RTCK clock. For all other cores it is recommended to start with a 10 MHz JTAG clock for boards that run in the 30MHz to 100MHz CPU clock range. For boards with CPU clock rates lower or higher you may change the JTAG clock to better suit your needs. Higher JTAG clock rates allow faster downloads.

Header

The Header section configures the JTAG header. Select the ARM header for all non-Texas Instruments devices and TMS320 for all OMAP and TMS470 devices. It is recommend not to change the remaining parameters – they are set to default ARM conditions.

Emulation

Emulation section allows to adjust the memory and reset timing. It is recommended to leave the Memory Access Speed on FAST. The Reset parameters are set to default values and can be adjusted as needed.

The JTAG configuration settings may be changed at any time during debugging from the **View/Target Configuration** menu.

Debug Handler Address (XScale only)

XScale processors, it is necessary to specify a 2KB memory region dedicated exclusively to the debug handler. This setting becomes the handler code's base address. It must be reserved strictly for this purpose, be 2KB in size and aligned to a

2KB address. The alignment requirements means that the address's 11 least significant bits must be zero.

11. The next step in custom target configuration is to select a Startup Macro. At this point a list of macro files will appear. These macros are for all target boards we know of. Obviously, if you were creating your own target configuration you will not find the macro on this list. However, if you have prepared your own macro ahead of time or it was given to you by someone else, please browse to the proper directory and select it here.

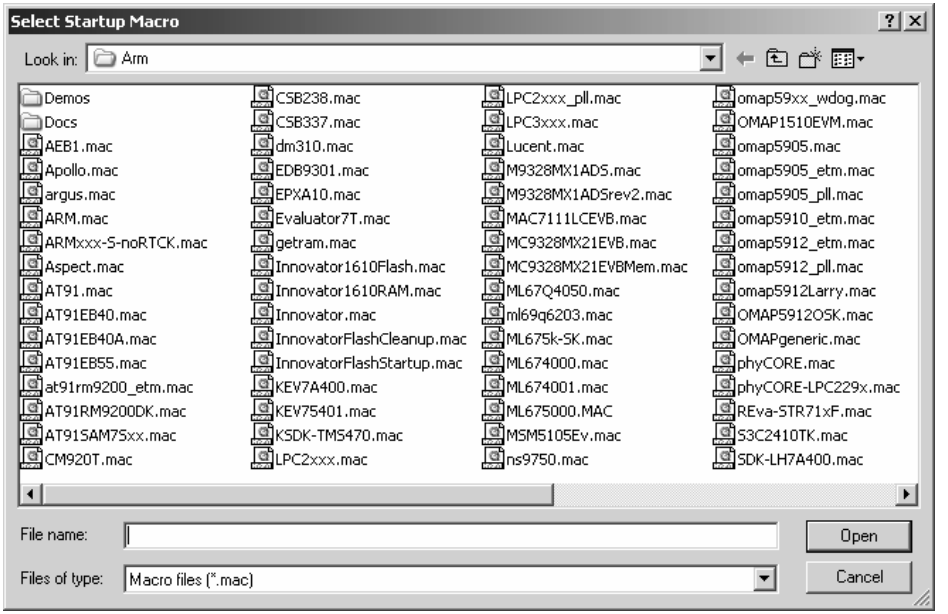

FIGURE 11 Selecting the startup macro file.

12. This concludes a custom target setup. After the debugger establishes connection with the CPU, the initial screen will look like this.

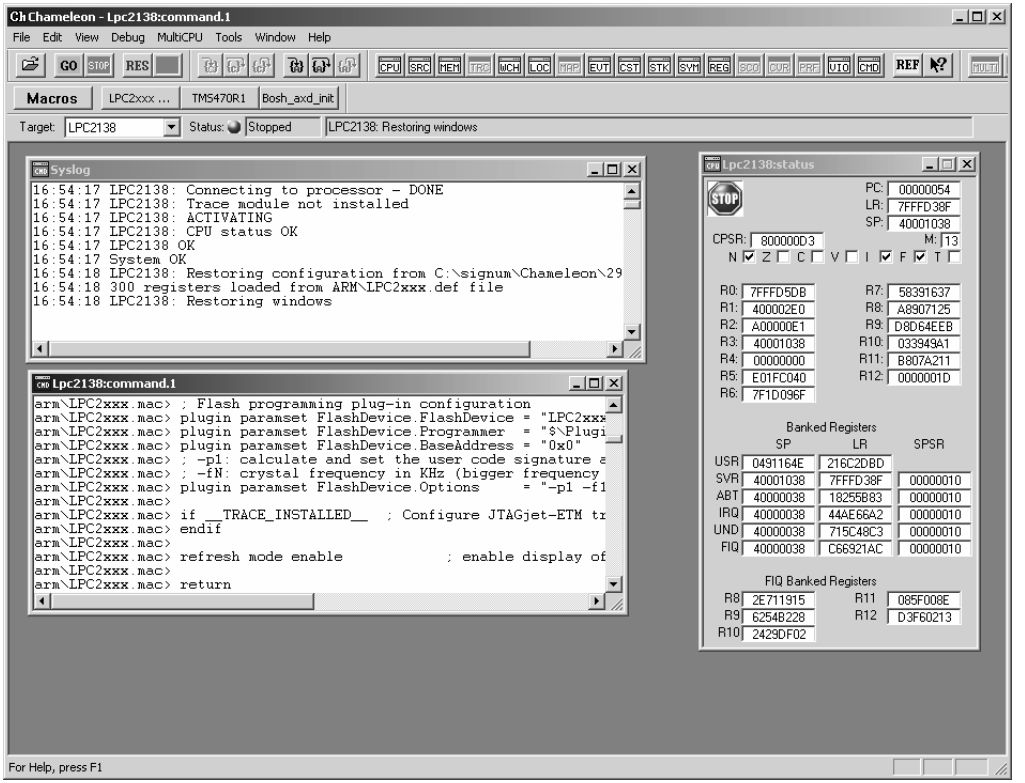

FIGURE 12 Chameleon's opening screen.

It is recommended that you familiarize yourself with the documents provided in the Chameleon Help menu before continuing with your project.

If you encounter difficulties when installing or starting the debugger, please provide the Signum Technical Support team with the .log and .ini files found in the Chameleon installation folder. For contact and additional information, please visit www.signum.com/support.htm.

❑