
CS1810xx / CS4961xx Bring-up Instructions

1. INTRODUCTION

CobraNet® firmware is stored in external FLASH memory which must be programmed before the CobraNet device can become operational.

Once an initial firmware load of the FLASH is accomplished, the device will be functional. All subsequent firmware updates can be accomplished via the network, using Cirrus-supplied tools such as CobraNet Discovery or the Python-based manufacturing script.

There are two means for accomplishing this initial firmware load:

- Use of pre-programmed FLASH
- Use of the CID debug interface tool

1.1 Pre-Programmed Flash Device

The simplest and most efficient approach, especially for volume production, is to pre-program the FLASH memory before it is soldered to the circuit board.

CobraNet firmware is delivered as a 1M byte binary file (example: cm18101_2_11_6.bin). The latest firmware may be downloaded from the Cirrus Logic web site at <http://www.cirrus.com>. The firmware binary file is an exact image of FLASH contents required to boot and operate the CobraNet chip.

A properly constructed CobraNet interface populated with a pre-programmed FLASH will, without intervention, boot and function properly from initial power-up.

It is recommended that the initial firmware image to be used is the cs18100_2_11.x.bin firmware file. This is the lowest common denominator firmware and will provide proper network functionality in ALL CobraNet-chip-based platforms. A manufacturer can then stock only one pre-programmed chip type and then, later, update specifically required firmware variants during the manufacturing process. FLASH will always need to be reprogrammed in any case as part of the MAC address assignment process.

1.2 I²C® Interface

The chip also features an I²C® interface. Use of the I²C port for device bring-up requires use of:

- I²C/Parallel interface adapter.
- Personal Computer
- CID software
- CobraNet Discovery Software

The software noted above is bundled with this document and can also be downloaded from the Cirrus Logic web site. The interface adapter can be constructed from the schematic included in this document or can be supplied by Cirrus Logic to qualified manufacturers.

Details on using the I²C interface and support software (CID) are provided in the “[I²C Interface and Software](#)” section.

Use of the I²C interface to load firmware is a two-step process. CID is used to load and run a base version of firmware in volatile RAM. Once running from RAM and operational on the network, either CobraNet Discovery or the Python manufacturing script must be used to permanently load the FLASH memory with the most current firmware version.

2. HARDWARE DIAGNOSTICS

The standard firmware includes a boot-up routine that performs a Power On Self Test (POST). POST tests internal memory, FLASH memory, the Ethernet controller, and audio clock circuitry. POST can discover and help diagnose many hardware design errors.

When booting from pre-programmed FLASH, POST errors are reported through blinking indicators on the Ethernet connectors. For detailed information on POST error reporting, see the **Error Reporting** section in the *CobraNet Programmer's Reference Manual*.

When booting from the I²C interface, POST errors are reported at the CID command line as described in the “[POST Results](#)” section on page 4.

3. I²C INTERFACE AND SOFTWARE

The I²C interface hardware connects between a standard bidirectional parallel printer port on a PC and the target system.

3.1 Software Installation

To install the software:

1. Unzip the contents of the software archive bundled with this document (CID_Distribution.zip) to an empty directory.
2. Open a command prompt window and change to the directory containing the software.
3. Run 'port95nt' to install the parallel port I/O driver.

3.2 Connecting the Target System

The I²C adaptor connects to a PC parallel printer port with a straight-through male-female 25-pin cable. The adaptor features a 10-pin header on the target side. The first four pins connect from the adaptor to the target. Pin 1 is marked with a square pad and is the pin furthest towards the bottom in [Figure 1](#).

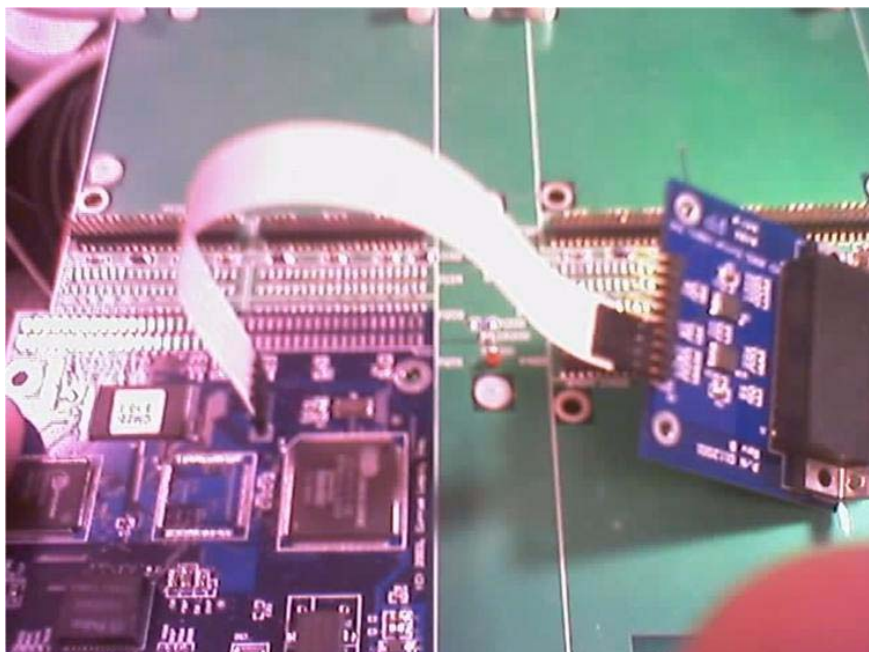


Figure 1. I²C Adaptor (right) Connected to an Example Target System (CM-2, left)

For target systems that do not use the standard I²C pin-out of the CM-2, I²C adaptor signals are described in [Table 1](#) and shown in the schematic included in the *CobraNet Hardware User's Guide*.

I ² C Adaptor Pin	I ² C Signal	CobraNet Chip Pin Number
1	Clock	4
2	Ground	Multiple (GND)
3	Data	3
4	3.3v	Multiple (VDDIO)

Table 1. I²C Signals

3.3 Starting the Target System

WARNING: CID will only function properly when code is not running on the target platform. It is designed to be used with an 'empty' FLASH. If attempting to use CID with a running device, the device's ability to boot must first be disabled and the device restarted. See ["Disabling the Boot Process" on page 4](#) if this step is required.

The following procedure will download CobraNet firmware to the target system's RAM and invoke execution.

1. From a command prompt at the directory where the CID software has been copied, run 'cid'
2. At the prompt, type "e lcm2.cmd". This command takes approximately 20 seconds to execute. Success is indicated by the following printout.

```

Stopped at 0x617d
Running from new location
data returned:
0x1ff0: 0x6000ff2b 0x00ff0000
x1: 0xffffffff
Stopped at 0x33<< - 0x33 at this line indicates success
Running from new location

```

The output above indicates that the interface is booted and functioning on the network. This step does not load firmware to FLASH memory. At this time, a base version of firmware has been loaded to volatile RAM, and the device is now running from RAM. CobraNet Discovery or the Python manufacturing script must next be used to connect to the device via the network and load the latest firmware version to the FLASH memory.

4. DEBUGGING THE HARDWARE

The download and boot process has been seen to fail with the following message in rare instances:

```

Stopped at 0x617d (or some other invalid address)
Running from new location
data returned:
0x1ff0: 0x6000ff2b 0x00ff0000
x1: 0x00000004
Stopped at 0x1f98(or some other invalid address)
Running from new location

```

The above example indicates a POST failure on the Ethernet MAC test. This is caused by an interaction between the CID software, the CobraNet chip, and the Ethernet controller. If you encounter a failure, retry the operation by resetting your hardware and then reissuing the "e lcm2.cmd". If success can be achieved after one or two tries, you can be assured that there is no problem with the CobraNet chip or with your hardware design.

5. MAC ADDRESS

The line `0x1ff0: 0x6000ff2b 0x00ff0000` shows the MAC address. The MAC address is stored in a section of the flash memory called the *configuration record*. The value shown here is a special default value used for the case when an un-programmed configuration record is detected.

In this case of empty FLASH, a special MAC address of `00:60:2b:ff:ff:00` is used.

The value should be viewed as consecutive words that are byte-order-reversed, with the last word (2 bytes) being of no significance. Units with default MAC addresses should not be used in an operational network until the configuration record is programmed with a valid MAC address.

MAC addresses must be globally unique. Blocks of Ethernet MAC addresses for production can be obtained directly from the IEEE or from Cirrus Logic.

The configuration record, with new MAC address, can be programmed by using the Cirrus-supplied Python manufacturing script in conjunction with the Cirrus-supplied PACNDisco and PACNFirm COM objects. These objects are automatically installed when CobraNet Discovery is installed.

If a valid MAC address has been previously programmed, the programmed MAC address will be shown. For example, a MAC address of `34:12:78:56:bc:9a` would be indicated by the following CID output:

`"0x1ff0: 0x12345678 0x9abcxxx"` – where 'xxx' can be anything.

6. POST RESULTS

The line, "stopped at 0x33", indicates POST has passed. If "stopped at 0x1f78" is shown instead, POST has failed and the failure code is indicated on the previous line, "x1: 0xffffffff". The value shown here (0xffffffff) is given in the case where no POST error was detected. Possible POST error codes are as follows.

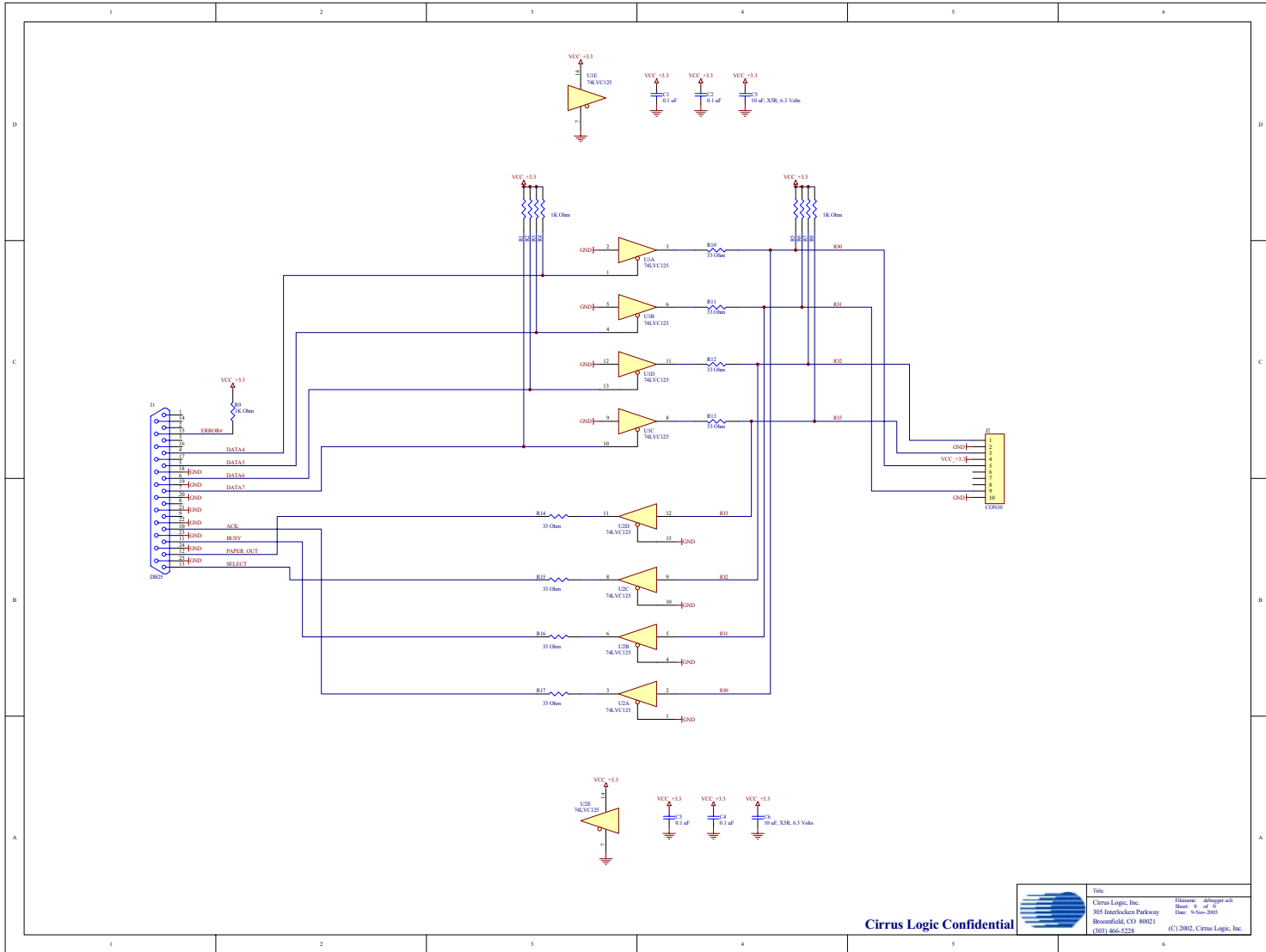
- x1: 0x00000004 - Ethernet controller test failed. Unable to access Ethernet controller. Problem with Ethernet controller or bus interface to Ethernet controller.
- X1: 0x00000007 - Firmware incompatible problem. Firmware is incompatible with CobraNet chip used.
- X1: 0x00000008 - Internal memory test failure. Problem with on-chip CS18101 memory.
- x1: 0x00000016 - Sample clock pull range test failed. The sample clock frequency could not be adequately adjusted. There is a problem with the VCXO or supporting circuitry.
- x1: 0x00000017 - Sample clock frequency test failed. The sample clock does not appear to be operating. There is a problem with the VCXO or supporting circuitry.

7. DISABLING THE BOOT PROCESS

There are two ways to prevent the target device from booting when the FLASH memory contains valid code.

- Ground the HS1 pin (pin 17) on the CobraNet chip. This will instruct the on-chip boot loader to skip loading firmware from FLASH memory.
- Invalidate the boot record in FLASH. Invalidation of the boot record will prevent all subsequent boots.

Invalidation of the boot record is accomplished by running the included 'eboot' script instead of the 'lcm2' script. This may require running multiple times before it is successful because CID will be conflicting with the existing firmware that is running. Reset the target device after each time 'eboot' is run.



Cirrus Logic Confidential

Title:	Cirrus Logic, Inc.	Filename:	Adapters.ich
	305 Interlocken Parkway	Sheet:	5 of 5
	Broomfield, CO 80021	Date:	9-Nov-2005
	(303) 466-5228		(C) 2002, Cirrus Logic, Inc.

Figure 2. CID Adapter



Contacting Cirrus Logic Support

For all product questions and inquiries contact a Cirrus Logic Sales Representative.
To find one nearest you go to <http://www.cirrus.com>

IMPORTANT NOTICE

Cirrus Logic, Inc. and its subsidiaries ("Cirrus") believe that the information contained in this document is accurate and reliable. However, the information is subject to change without notice and is provided "AS IS" without warranty of any kind (express or implied). Customers are advised to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, indemnification, and limitation of liability. No responsibility is assumed by Cirrus for the use of this information, including use of this information as the basis for manufacture or sale of any items, or for infringement of patents or other rights of third parties. This document is the property of Cirrus and by furnishing this information, Cirrus grants no license, express or implied under any patents, mask work rights, copyrights, trademarks, trade secrets or other intellectual property rights. Cirrus owns the copyrights associated with the information contained herein and gives consent for copies to be made of the information only for use within your organization with respect to Cirrus integrated circuits or other products of Cirrus. This consent does not extend to other copying such as copying for general distribution, advertising or promotional purposes, or for creating any work for resale.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). CIRRUS PRODUCTS ARE NOT DESIGNED, AUTHORIZED OR WARRANTED FOR USE IN PRODUCTS SURGICALLY IMPLANTED INTO THE BODY, AUTOMOTIVE SAFETY OR SECURITY DEVICES, LIFE SUPPORT PRODUCTS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF CIRRUS PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK AND CIRRUS DISCLAIMS AND MAKES NO WARRANTY, EXPRESS, STATUTORY OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE, WITH REGARD TO ANY CIRRUS PRODUCT THAT IS USED IN SUCH A MANNER. IF THE CUSTOMER OR CUSTOMER'S CUSTOMER USES OR PERMITS THE USE OF CIRRUS PRODUCTS IN CRITICAL APPLICATIONS, CUSTOMER AGREES, BY SUCH USE, TO FULLY INDEMNIFY CIRRUS, ITS OFFICERS, DIRECTORS, EMPLOYEES, DISTRIBUTORS AND OTHER AGENTS FROM ANY AND ALL LIABILITY, INCLUDING ATTORNEYS' FEES AND COSTS, THAT MAY RESULT FROM OR ARISE IN CONNECTION WITH THESE USES.

Cirrus Logic, Cirrus, the Cirrus Logic logo designs, and CobraNet are trademarks of Cirrus Logic, Inc. All other brand and product names in this document may be trademarks or service marks of their respective owners.

I²C is a registered trademark of Philips Semiconductor Corporation.