CIRRUS LOGIC®

CDB89712

User's Manual

CDB89712 HARDWARE AND SOFTWARE USER'S MANUAL

ARM®

This document describes the CS89712 Development Board provided by Cirrus Logic Inc. No warranty is given for the suitability of this design for any purpose other than demonstrating the functional operation of the CS89712. The information contained in this document is subject to change without notice.

P.O. Box 17847, Austin, Texas 78760
(512) 445 7222   FAX: (512) 445 7581
http://www.cirrus.com

TABLE OF CONTENTS

Contacting Cirrus Logic Support

For a complete listing of Direct Sales, Distributor, and Sales Representative contacts, visit the Cirrus Logic web site at:
http://www.cirrus.com/corporate/contacts/

Preliminary product information describes products which are in production, but for which full characterization data is not yet available. Advance product information describes products which are in development and subject to development changes. Cirrus Logic, Inc. has made best efforts to ensure 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). No responsibility is assumed by Cirrus Logic, Inc. for the use of this information, nor for infringements of patents or other rights of third parties. This document is the property of Cirrus Logic, Inc. and implies no license under patents, copyrights, trademarks, or trade secrets. No part of this publication may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc. Items from any Cirrus Logic website or disk may be printed for use by the user. However, no part of the printout or electronic files may be copied, reproduced, stored in a retrieval system, or transmitted, in any form or by any means (electronic, mechanical, photographic, or otherwise) without the prior written consent of Cirrus Logic, Inc.Furthermore, no part of this publication may be used as a basis for manufacture or sale of any items without the prior written consent of Cirrus Logic, Inc. The names of products of Cirrus Logic, Inc. or other vendors and suppliers appearing in this document may be trademarks or service marks of their respective owners which may be registered in some jurisdictions. A list of Cirrus Logic, Inc. trademarks and service marks can be found at http://www.cirrus.com.

LIST OF FIGURES

LIST OF TABLES

Chapter 1: The CS89712 Development Kit

1.1 DESCRIPTION OF THE CDB89712 DEVELOPMENT KIT

The CS89712 development kit, which is part of the ARM product family, is intended for system engineers who want to develop products based upon the CS89712 ultra-low power system-on-chip (SOC). The kit can also be useful for software developers writing device drivers and applications for the CS89712.

1.2 ORDERING THE CDB89712 DEVELOPMENT KIT

You can order the CDB89712 Development Kit from your local Cirrus Sales Representative using the following part number:

- CS89712 Development Kit: CDB89712

1.3 KIT CONTENTS

The CDB89712 Development Kit contains the following:

- Cirrus Logic CS89712 Development Board - CDB89712
- Cirrus Logic CS89712 Development Kit CD
  - CS89712 Data Sheet
  - Component Data Sheet
  - Orcad Schematic
  - PDF Schematic
  - Sample Software Applications
  - ARM SDT 2.50 60 day evaluation version
- Green Hills MULTI 2000 - 30 day evaluation version
  - ThreadX Operating System demo
  - Interniche Web Server demo
  - Interniche TCP/IP demo

CIRRUS LOGIC®
P.O. Box 17847, Austin, Texas 78760
(512) 445 7222 FAX: (512) 445 7581
http://www.cirrus.com

- Sample Software Applications
- LynuxWorks' BlueCat Enhanced Tool Kit for Windows Host and the CS89712 Linux Board Support Package
  - Sample Software Applications
- VxWorks Board Support Package from WindRiver
- Macraigor OCD ("Wiggler") Debugging Tool
- Ice_boot Debug Monitor Preloaded in FLASH Memory
- Serial Cable
- AC Adapter

1.4 SOFTWARE DEVELOPMENT TOOLKITS PORTED TO THE CS89712

Software developers can choose from a wide range of software development tools to produce software for the CS89712 Development Kit. The CS89712 Development Kit supports the following software development toolkits:

Green Hills
• SOFTWARE, INC. •

- MULTI 2000 Integrated Development Environment from Green Hills Software - The CS89712 Development Kit includes a 30-day evaluation version of the MULTI Integrated Development Environment toolkit is a development environment for embedded applications using C,C++, Embedded C++, Ada 95, or FORTRAN languages. For more information about the MULTI development environment see:

  www. ghs.com/products/MULTI_IDE.html

  To upgrade to a fully licensed version of the MULTI Integrated Development environment, contact Green Hills at www.ghs.com/html/contact.htm.

ARM®

- ARM® Software Developers Toolkit, Ver. 2.50 — The CS89712 Development Kit includes a 60-day evaluation version of the ARM Toolkit. The ARM Toolkit includes a compiler, assembler, linker, and debugger.

For a complete list of development tools supporting the ARM architecture refer to:

http://www.arm.com/DevSupp/SoftwDevTools

To upgrade to a full- licensed version of the ARM Software Developers Toolkit, contact ARM at:

http://www.arm.com/DevSupp/ordering.html

1

- GNU Pro ETS (Embedded ToolSuite) from Red Hat®, Inc., a product formerly sold by Cygnus Solutions, is a set of software development tools based on the open source GNU standard. The CS89712 Developers Toolkit does not include an evaluation copy of this software development environment. To learn more about GNU Pro ETS and about how to order this product, go to this Web site:

www.cygnus.com/gnupro/gnupro_ets.html

BlueCat Enhanced Development Tool Kit distributed by LynuxWorks.

BlueCat Linux is an open source distribution of Linux version 2.2.12, adapted to meet the requirements of embedded developers. BlueCat Linux is enhanced with cross development and embedding tools and includes components that run on the development system (the host), and on the embedded system (target). It also contains demonstration programs and test suites to help the developer validate Linux-based applications. The CS89712 Development Kit includes a full version of the BlueCat Enhanced Development Tool Kit for Windows Host. The BlueCat for Linux host is also available. Contact www.lynuxworks.com for more information.

1.5 CS89712 DEVELOPMENT KIT BOARD

The development kit board is designed as a prototype. It has headers to access the various I/O ports of the CS89712, and for an interface to a logic analyzer.

1.6 MAIN FEATURE SET

The development board has the following features:

- Cirrus Logic CDB89712 Evaluation Board
- Microprocessor
  - CS89712 processor running at up to 73.728 MHz
- Memory
  - DRAM:16 Mbytes of SDRAM.
  - FLASH: A total of 8 Mbytes.
- Peripherals
  - Power Management Switches
  - Two Serial Ports, primary Serial interface switchable between RS-232 and IrDA
  - RJ45 Jack for on-chip Ethernet
  - Logic Analyzer Connectors
  - General Purpose I/O Expansion Connector
  - Prototyping Area

Chapter 2: Overview

2.1 SCOPE OF THIS MANUAL

This document describes the board-level hardware and software components that comprise the CDB89712 Cirrus Logic Development Board. This document points the user to vendor documents / Internet sites where the reader can find more specific details about other processors and devices (ICs) used on the CDB89712 development board. The scope of this manual is limited to describing the printed circuit board, connections between components on that board, and between the board and other components intended to connect to the evaluation board.

2.2 REVISION LEVELS ASSOCIATED WITH THIS MANUAL

This document assumes you are using following revision levels when you use this manual:

- Your development board is Revision B of the CDB89712 board.
- The software preloaded onto the CDB89712 Development Toolkit is the Ice_boot software. Ice_boot initializes the SDRAM, MMU and cache memory. It is used in conjunction with Green Hills Multi and the Wiggler that come with the CDB89712, or with the Multi_ICE tool from ARM, Ltd.

2.3 DOCUMENT CONVENTIONS

2.3.1 Terms and Definitions

Table 2-1 contains the acronyms, abbreviations, and terms used in this User's Manual.

Acronym / Abbreviation	Definition
ADC	Analog-to-Digital Converter
Angel	ARM Toolkit target debug monitor software
ARM	Advanced RISC Machines. This is the architecture of the microprocessor at the heart of the CS89712.
CODEC	Coder/decoder (usually a combination of ADC and DAC).
DAC	Digital to Analog Converter
DAI	Digital Audio Interface
FIFO	First In First Out.
FLASH	Electrically Erasable Read Only Memory
GUI	Graphical User Interface.
I^2C	Stands for "I-squared-C" which is a low-speed serial communication standard
I^2S	Stands for "I-squared-S" which is a serial audio data communication standard
Ice_boot	Program that initializes SDRAM interface, MMU and cache. Preloaded on CDB89712 Flash memory. Used with Green Hills Multi and Wiggler.
IrDA	Infrared Data Association
JTAG	Joint Test Action Group
LCD	Liquid Crystal Display
MMU	Memory Management Unit.
NAND	Not And (opposite of logical And)
PWM	Pulse Width Modulation
RTC	Real-Time-Clock. This clock maintains time/date information, and in the case of the CS89712 is Y2K compliant.
SDRAM	Synchronous Dynamic Random Access Memory. Only kind of DRAM that can be used with CS89712.
Smart Media	Alternate name for SSFDC
SSFDC	Solid State Floppy Disk Card, a standard for NAND Flash based non-volatile storage cards.
TPE	Twisted Pair Ethernet
UART	Universal Asynchronous Receiver and Transmitter
USB	Universal Serial Bus. This is a high-speed serial interface intended for interfacing scanners, printers, joysticks and other devices to a PC

Table 2-1. Terms and Definitions

2.3.2 Units of Measurement

Table 2-2 describes the units of measurement used in this manual.

Symbol	Unit of Measure
bpp	Bits per pixel. The number is bits used in a display.
bps	Bits per second. Used to describe the transfer rate of a serial interface.
°C	degree Celsius
Hz	hertz (cycle per second)
kbps	kilobits per second
kbyte	kilobyte (1,024 bytes)
kHz	kilohertz
kΩ	kilohm
Mbps	megabits (1,048,576 bits) per second
Mbyte	megabyte (1,048,576 bytes)
MHz	megahertz (1,000 kilohertz)
µA	microampere
µF	microfarad
µW	microwatt
µs	microsecond (1,000 nanoseconds)
mA	milliampere
mW	milliwatt
ms	millisecond (1,000 microseconds)
ns	nanosecond
V	volt
W	watt

Table 2-2. Units of Measurement

2.3.3 General Conventions

Hexadecimal numbers are presented with all letters in uppercase and a lowercase ‘h’ appended or with a 0x at the beginning. For example, 0x14 and 03CAh are hexadecimal numbers. Binary numbers are enclosed in single quotation marks when in text (for example, ‘11’ designates a binary number). Numbers not indicated by an ‘h’, 0x or quotation marks are decimal.

Registers are referred to by acronym, as listed in the tables on the previous page, with bits listed in brackets MSB-to-LSB separated by a colon (:) (for example, CODR[7:0]), or LSB-to-MSB separated by a hyphen (for example, CODR[0–2]).

The use of “tbd” indicates values that are “to be determined” “n/a” designates “not available,” and “n/c” indicates a pin that is a “no connect.”

2.4 ENSURING THAT YOU HAVE THE LATEST REVISION OF THIS DOCUMENT

This document describes the hardware and software components of the CDB89712-DK product. Any changes to the Development Kit will be described in future revisions of this manual and in Errata Sheets, if any, that accompany this document. The latest version of this manual, as well as documentation associated with all the other products from Cirrus Logic can be found on the Cirrus Internet site at:

http://www.cirrus.com/

2.5 SOURCES OF ADDITIONAL INFORMATION

- Internet addresses for suppliers of parts used in the CDB89712 Development Kit:
- 3M http://www.mmm.com/interconnects
- ARM: http://www.arm.com
- AMP: http://connect.amp.com
- Bourns: http://www.bourns.com
- Coilcraft: http://www.coicraft.com
- Halo Electronics: http://www.haloelectronics.com
- Intel: http://www.intel.com/developer
- Kemet: http://www.kemet.com
- LynuxWorks: http://www.lynuxworks.com
- Maxim: http://www.maxim-ic.com
- Micrel: http://www.micrel.com
- Microchip: http://www.microchip.com
- NEC: http://www.necel.com
- Sipex: http://sipex.com
- Vishay: http://www.vishay.com

2.5.1 Style Conventions

This manual follows these style conventions:

- The first number in the section number that precedes each heading equals the number of the chapter in which the heading appears. For example, the heading, “2.5 Sources of Additional Information” is found in Chapter 2.
- Table and figure numbers begin with the chapter number; for example, Figure 2-1 and Table 2-2 would be in chapter 2.

Chapter 3: Details of Hardware Design

3.1 INTRODUCTION

This chapter describes the major components on the CS89712 Development Board. A complete CS89712 data sheet can be downloaded from the Cirrus Logic website at
http://www.cirrus.com.

3.2 CS89712 PROCESSOR

The CS89712 is an ARM 32-bit RISC microcontroller incorporating a wide range of on-chip peripherals, including an on-chip Ethernet controller. The core of the CS89712 is designed with an ARM720T processing unit comprising an ARM7 microprocessor, enhanced MMU and 8 Kbytes of 4-way set-associative cache. The CS89712 is a 3.3V (core processor voltage is 2.5V) device with a maximum input clock of 73.728Mhz, achieving a performance level equivalent to 65 MIPS with a maximum power consumption of 170mW. An internal functional block diagram is shown in Figure 3-1.

3.3 FUNCTIONAL BLOCK DIAGRAM

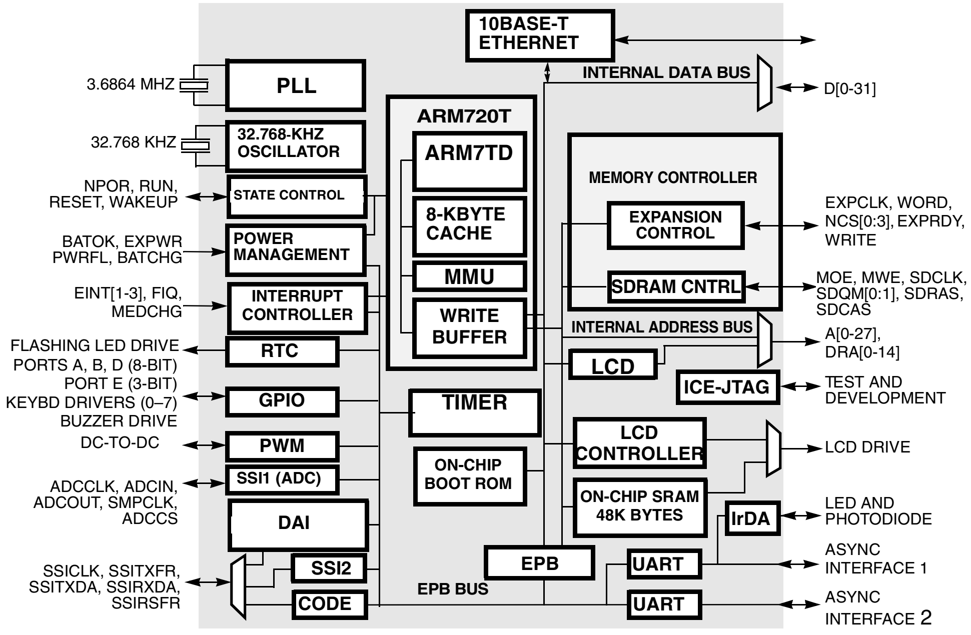

Figure 3-1. CS89712 Functional Block Diagram

3.4 MEMORY

There are two types of external memory present on the development board:

- 16MB SDRAM
- 8MB FLASH

3.4.1 SDRAM

The SDRAM provided is sufficiently large to meet the needs of all potential applications and provides a fast access data area.

The 16 MByte SDRAM memory is provided as a single 32-bit bank, made up from two 1,048,576 X 16 X 4 (word X bit X bank) 64Mbit memories.

When the board is initialized from Ice_boot in the Flash memory, the SDRAM is mapped to 0x0000:0000 to 0x007F:FFFF. This SDRAM area is used for code/data storage, and also contains the LCD buffer data.

It may be desirable to rearrange the SDRAM. This can be accomplished by programming the MMU. A detailed discussion of setting up the MMU is beyond the scope of this manual, and the reader is directed to the *ARM Architectural Reference Manua*l for further information.

3.4.2 FLASH Memory

The FLASH memory is provided in one bank, made up of two 32 Mbit devices, arranged as one contiguous block of 8 Mbytes. Accesses to FLASH are 32 bits wide.

The devices used on the development board are Intel TE28F320B3BA110.

Each block may be erased up to 100,000 times before end of life. If frequent writes to FLASH are likely, the life of the FLASH memory can be extended by distributing data storage over the entire addressable space of the FLASH memory.

The FLASH devices have 110nS access time. When running at 73.728MHz, a minimum of 5 wait-states is required to reliably communicate with the FLASH memory.

Details on erasing, writing, locking and unlocking the sectors of the FLASH devices can be found in the Intel data sheet for the L28F320 FLASH devices, Intel Order Number 290645-009. A reference datasheet is provided on the CD, l28f320.pdf.

3.5 CS89712 CLOCKS

The CS89712 requires two clocks for normal operation:

- The main system clock (default clock)
- The real time clock

The real-time clock oscillates at a frequency of 32.768 KHz. This clock provides the time base for the integrated real time clock and is provided by a simple tuning fork type crystal.

The processor derives its clock from a 3.6864 MHz crystal connected to the master oscillator pins. 73.728 MHz is the maximum operating frequency of the CS89712.

3.6 POWER MANAGEMENT MODES

The power management modes implemented by the CS89712 enable complete control of system power consumption. While the development board does not actually implement power management features, it does provide access to the CS89712's power management signals. This allows power management functionality to be tested and verified.

The four signals that control the power management of the CS89712 are:

- nEXTPWR
- nPWRFL
- BATOK
- nBATCHG

Four switches positioned at the edge of the board control these signals. These switches allow various transitions to be made between the operating states of the CS89712. Figure 4-13 shows the switch state for each of the following signals.

3.6.1 Description of Power Management Signals

Here is a brief explanation of the four power management signals.

3.6.1.1. BATOK

When asserted indicates a battery failure condition. A falling edge on the BATOK signal generates a FIQ. This condition would initiate a transition to Standby State.

The CS89712 will only transition to the Operating State if both BATOK and nPWRFL are returned to their negated states.

3.6.1.2. nBATCHG

When asserted indicates a "no battery" condition, or battery voltage has fallen below the minimum operating threshold.

3.6.1.3. nPWRFL

When asserted indicates a power fail condition. This condition will force a transition to the Standby State.

3.6.1.4. nEXTPWR

This signal must be asserted when external power is applied. Placing this switch in the asserted state will cause a transition to the Standby State. Operating State is resumed when nPWRFL is negated. (This is a simulation of the fail-safe condition that prevents the CS89712 from operating if the power supply or battery is inadequate.)

3.7 POWER STATES

The CS89712 supports three power states:

- Operating
- Standby
- Idle

3.7.1 Description Of Supported Power States

Figure 3-2, "CS89712 Operating States," details the operating states of the CS89712 microcontroller. Table 4-11, "Power States of the CS89712's Peripherals by Operating State," lists the status of the CS89712's peripherals in the different operating states.

3.7.1.1. OPERATING

All functions and integrated peripherals of the CS89712 are running.

3.7.1.2. STANDBY

Standby state effectively turns off the CS89712. The 73.728MHz PLL is turned off, there is no display and the internal peripherals are off. The RTC is active, and all system states and memory contents are maintained (SDRAM is placed in self-refresh mode).

The RUN signal is driven low.

Standby state is entered when power is applied, or a system reset is activated.

3.7.1.3. IDLE

The Idle state is similar to the Operating state except that the processor clock is turned off (PLL is still active).

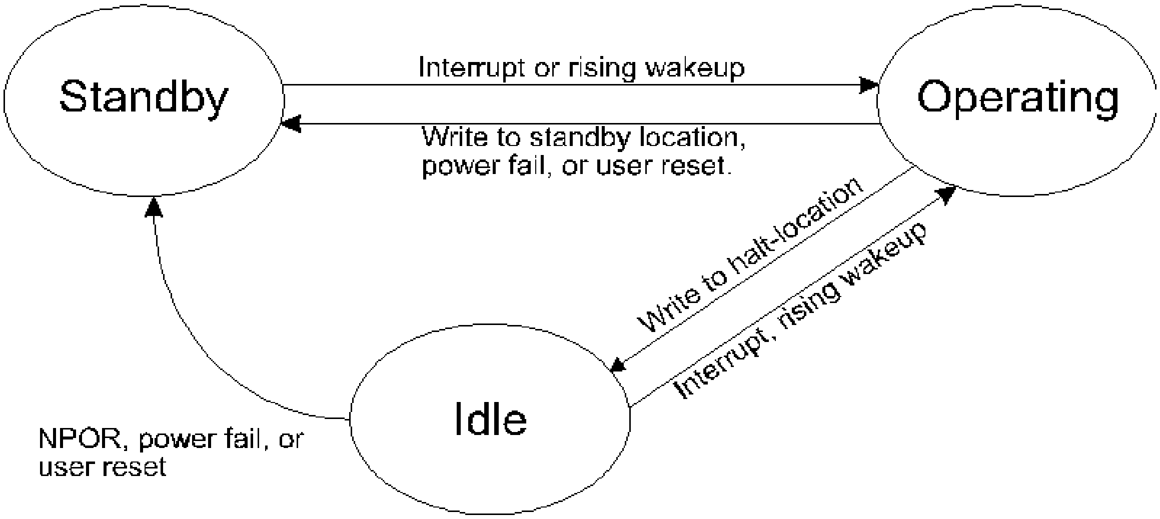

Figure 3-2. CS89712 Operating States

3.8 RESETTING /WAKING UP THE DEVELOPMENT BOARD

There are several momentary signals that are used to either reset or wakeup the development board.

3.8.0.1. Ethernet Port power state

The Ethernet port has a software initiated power down state that is independent of the ARM core.

3.8.1 Resetting the Board

There are three asynchronous reset signals supported by the CS89712:

- nPOR
- nURESET
- nPWRFL.

Here is a brief description of the reset signals:

3.8.1.1. nPOR

This signal is the initial reset signal and is only activated when power is first applied to the board. The development board implements this signal with a simple resistor and capacitor network with a time constant of approximately 10mS. In addition, a momentary switch labeled POR is available to implement control of this signal. Note that a power-on-reset will also reset the RTC.

3.8.1.2. nURESET

User reset; this input resets the ARM processor without resetting many of the registers initalized by nPOR. (See the CS89712 data sheet for a complete description.) nURESET allows the system to be initialized from an external source, such as a watchdog timer or the user. On the CDB89712, nURESET is activated with a momentary switch labeled RESET located at the front of the board.

3.8.1.3. nPWFRL

Power failure; when active, this signal will generate an internal reset and place the CS89712 into Standby State. This signal is generated from a switch labelled PWRFL on the CDB89712.

3.8.2 Waking up the Board

Once the CS89712 has been reset, it enters Standby. The transition from the Standby state to the Operating state is triggered by a rising edge on the Wakeup pin. Control of the Wakeup pin is implemented with a momentary switch labelled WAKEUP located at the front of the development board. For more information on resetting and waking up the ARM processor, see Application Note AN186, “Bringing Up the EP72/73XX Device”, which is available on the Web site.

3.9 DEVELOPMENT BOARD INTERFACES

3.9.1 Serial Ports

There are two serial ports on the board, numbered 0 and 1. Serial Port 0 is always in its powered up state. Port 0 can be disabled, but not powered down, by installing a jumper on JP36, which holds the Data Terminal Ready (DTR) line low. If the jumper is removed, the DTR line is held high. Serial Port 1 is disabled by installing a jumper on JP35.

3.9.2 Ethernet Port

The CDB89712 has one 10BASE-T Ethernet port, J3. It is connected to the Ethernet pins on the CS89712 through a set of transformers in U11.

3.9.3 JTAG Connectors for ICE

The two JTAG connectors allow a 14 pin or 20 pin cable to be connected as an In Circuit Emulator.

3.9.4 IrDA Port

U14 is an IrDA transceiver that connects directly to the IrDA port on the CS89712.

3.9.5 Expansion Headers

The I/O pins of the CS89712 are brought out to headers for connecting a display, keyboard, Digital Audio Interface (DAI) and other I/O devices. These devices were not included on the CDB89712 so that the board could be customized by the user through the expansion headers.

Chapter 4: Set-Up and Operation Reference

4.1 INTRODUCTION

This chapter describes how to establish connectivity between your host PC and the CS89712 Development Board. The other sections in this manual serve as a reference chapter when you want to know the configuration settings for controlling the various components of the development board.

4.2 CONNECTING THE HOST PC TO THE DEVELOPMENT BOARD

There are two methods for connecting a host PC to the development board to debug an application:

- Connect your host PC to the JTAG interface connector via a cable and port specified by the debugging software you choose to use. The Wiggler supplied with the board connects to the PC parallel port and to the 14 pin JTAG header JP21. Pin 1 of the ribbon cable connector (with the red stripe on the cable) goes towards the edge of the board - that is, towards the ground lug JP1. Figure 4-1, "Wiggler Connected to CDB89712," shows the Wiggler connected to the CDB89712.

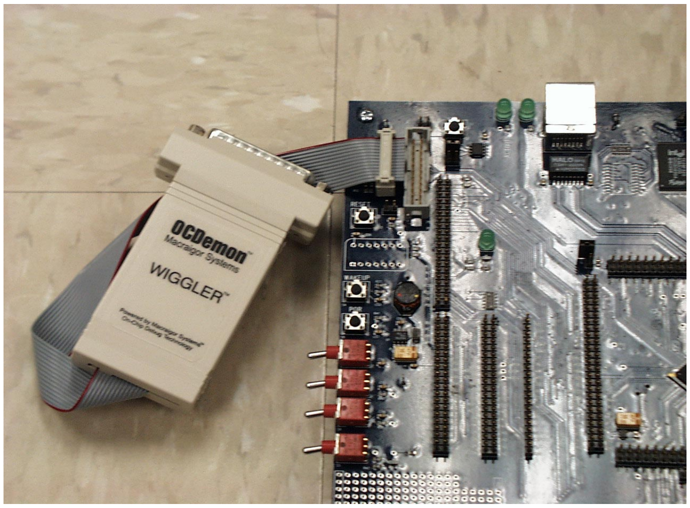

Figure 4-1. Wiggler Connected to CDB89712

- Connect a null modem cable, which is supplied with your CS89712 Development Kit, from a host PC to the Development Board. Communication is enabled by connecting UART1 (Serial Port 0) on the development board to either COM1 or COM2 of the host PC. See “Angel Debug Monitor” on page 2 for more information about the Angel debug monitor software.

The pin connections for the null modem connector are shown in Figure 4-2.

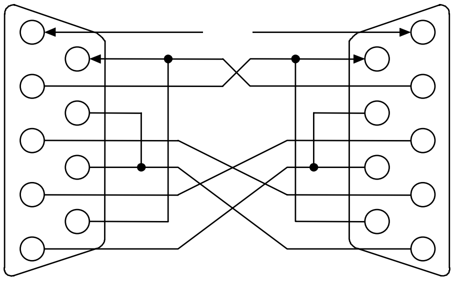

Figure 4-2. Null Modem Cable Wiring

4.3 CS89712 DEVELOPMENT BOARD LAYOUT

Figure 4-3 shows the layout of the CDB89712, Rev. A and Rev. B:

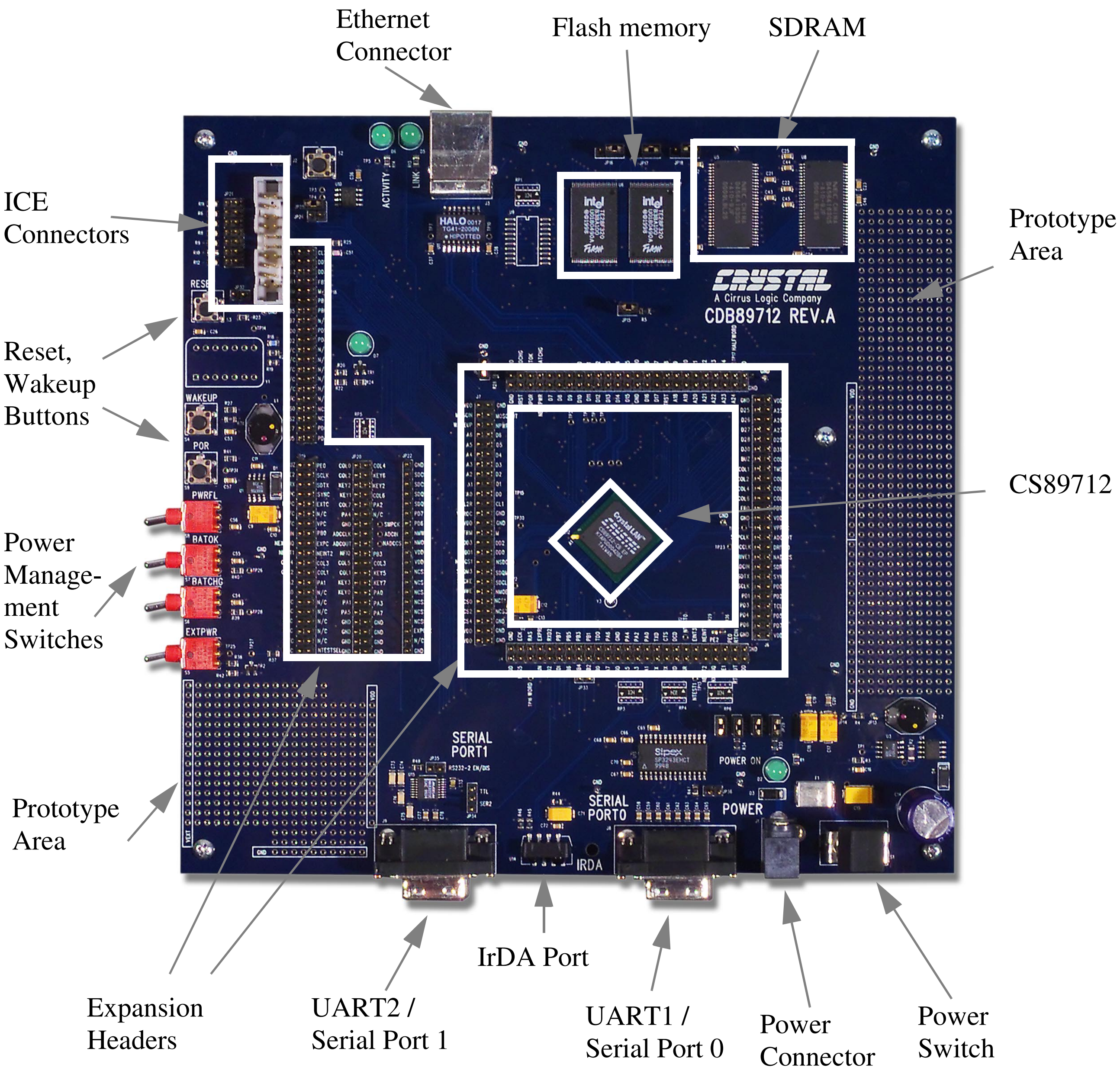

Figure 4-3. Layout of CS89712 Development Board with Major Board Components Identified

4.4 DEVELOPMENT BOARD JUMPER AND SWITCH SETTINGS

This section describes the jumpers switches, indicators and other signals that are available on the CS89712 Development Board. For more information about the functions they control, see the *CS89712 Data Sheet*.

4.4.1 Controlling / Changing Processor States

Table 4-1 shows the jumper settings for controlling and changing the operating states of CS89712 microcontroller.

NOTE: Each CS89712 Development Board is checked before it leaves the factory to make sure it powers up correctly. All jumpers are set to allow you to power up the board.

Jumper	Signal	Description	
JP31	MEDCHG	● ●	Boot in normal mode, start executing from flash, location 0x0000:0000. (Factory setting)
		○ ○	Boot from internal boot-ROM, reverse chip select mapping & begin executing from what is normally expansion page 7, now at location 0x0000:0000.
JP35	UART2 EN/DIS	● ●	Enable UART2 line driver. (Factory setting)
		○ ○	Disable UART2 line driver.
JP27/28	PE[1:0]	○ ○ ○ ○	32-bit boot operation. (Factory setting)
		○ ● ○ ●	8-bit boot operation.
		● ○ ● ○	16-bit boot operation.
		● ● ● ●	Invalid.
JP32	nURESET	● ●	nURESET pulled high. (Factory setting)
		○ ○	nURESET pulled low (S3 disabled).
JP29/30	nTEST[1:0]	○ ○ ○ ○	Factory setting shown. Please consult the CS89712 *User's Manual* for further information on these control lines. [a]
J2		Factory test only.	
JP25	EE Data In	○ ○	Enables Ethernet EEPROM.
JP26		Factory test only.	

Table 4-1. Jumper settings Used to Enable Certain Processor States

JP33		Factory test only.
JP34	TxD2 RxD2	Shunt pin 1 and 2 to loopback UART 2.

Table 4-1. Jumper settings Used to Enable Certain Processor States (Continued)

[a] You will need to jumper JP29 and 30 when you use JTAG or Multi-ICE debugging.

4.4.2 Flash Memory Interface Settings (JP15-18)

The interface to the Flash memory is controlled by jumpers JP15, 16, 17 and 18. The settings are shown in table Table 4-2.

Jumper	Name	Description
JP15	16 / 32 BIT FLASH ENABLE	● ● 16 bit Flash memory enabled. disables the most significant word of the Flash memory (U7).
		○ ○ 32 bit Flash memory enabled. Enables the most significant word of the memory (U7). (Factory setting)
JP16, JP17, JP18	SELECT 16 / 32 BIT FLASH	●|○ ○ JPn 16 bit selected. This adjusts the addressing of the Flash memory to short word (16 bit) addressing, so that A1 is the LSB of the address going to the Flash memory. This setting is only valid when JP15 is not installed.
		○ ○|● JPn 32 bit selected. This setting aligns the Flash memory address to long word (32 bit) boundaries, so that the LSB of the address to the Flash memory is A2. This setting is only valid when JP15 is installed. (Factory setting)

Table 4-2. Flash Memory Interface Settings

4.4.3 Momentary Switches

The switches listed in Table 4-3 are switches that are held down momentarily to initiate an action on the development board.

Switch Label	CS89712 Signal	Switch Description
WAKEUP	WAKEUP	Forces CS89712 into Operating state
RESET	nURESET	Resets CS89712, CS8900A, PS6700, and JTAG/ICE.
POR	nPOR	Power on reset. This switch can override any processor state, and will reset the real-time-clock.
S2		Factory test only.

Table 4-3. Description of Momentary Switches

4.4.4 LED Indicators

Table 4-4 describes the activity displayed by the LED indicators on the CS89712 Development Board.

Location	Name	Description
D7	DIAG	User Controllable Diagnostic Indicator
D5	LINK	Valid Link Detected on LAN
D6	ACTIVITY	Activity Detected on LAN
D2	POWER	Board Power On

Table 4-4. LED Indicators

4.5 SYSTEM ASSIGNMENTS

4.5.1 Memory Map

Physical Address Space	Description	Decode
0x0000:0000 – 0x00FF:FFFF	16 Mbytes SDRAM	
0x0000:0000 - 0x0000:4000	Ice_boot program	
0x0000:4000 - 0x0000:8000	Page Table for MMU	
0x0000:8000 – 0x00FF:FFFF	User Program Space	
0x0100:0000 – 0x017F:FFFF	8 Mbytes FLASH BANK 1	nCS0
0x0180:0000 – 0x01FF:FFFF	8 Mbytes FLASH BANK 0	nCS0
0x1000:0000 – 0x1FFF:FFFF	Reserved	nCS0
0x2000:0000 – 0x2FFF:FFFF	On Board Ethernet Port	nCS2
0X3000:0000 – 0X3FFF:FFFF	40 KBytes Expansion Header: Parallel port, keyboard	nCS1
0x4000:0000 – 0x4FFF:FFFF	Expansion Header: USB Controller	
0x5000:0000 – 0x5FFF:FFFF	Expansion Header	
0x6000:0000 – 0x6000:BFFF	48 KBytes On chip SRAM	
0x7000:0000 – 0x7000:007F	On Chip Boot Code, 128 Bytes	
0x8000:0000 – 0x8000:02FF	CS89712 Internal Registers	

Table 4-5. Memory Map for the CS89712 Microcontroller, Booting Ice_boot

Physical Address Space	Description	Decode
0x0000:0000 – 0x0000:FFFF	Flash Bank, ROM Space	nCS0
0x0001:0000 – 0x007F:FFFF	Flash Bank	nCS0
0x2000:0000 – 0x2000:02FF	Expansion Header	nCS2
0x2000:0300 – 0x2000:030F	Ethernet Port	nCS2
0X2000:0310 – 0X2FFF:FFFF	Expansion Header	nCS2
0x3000:0000 – 0x3FFFF:FFFF	Expansion Header	nCS3
0x4000:0000 – 0x4FFF:FFFF	Expansion Header	nCS4
0x5000:0000 – 0x5FFF:FFFF	Expansion Header	nCS5
0x6000:0000 – 0x6000:BFFF	On Chip SRAM, 48 KBytes	nCS6
0x7000:0000 – 0x7000:007F	On Chip Boot Code, 128 Bytes	nCS7
0x8000:0000 – 0x8000:2600	CS89712 Registers	
0xC000:0000 – 0xC001:FFFF	SDRAM, LCD Buffer	
0xC002:0000 – 0xC002:7FFF	SDRAM, Work Area	
0x0802:8000 – 0x08FF:FFFF	User Program Space SDRAM	

Table 4-6. Memory Map for the CS89712, Booting Angel

4.5.2 GPIO Assignments

Port	Reset State	I/O	Description
PA[7:0]	Input	I	Keyboard Row Data JP20
PB0	Input	I/O	Available for general use on JP19
PB1	Input	O	RTS control line for UART0
PB2	Input	O	Ring Indicate line for UART0
PB3	Input	O	Available for general use.
PB4	Input	O	Available for general use.
PB5	Input	O	Available for general use.
PB6	Input	O	Available for general use.
PB7	Input	O	Available for general use.
PD0	Output – low	O	Diagnostic LED control line.
PD1	I/O	O	Available for general use.
PD2	Output – low	O	Available for general use.
PD3	Output – low	O	Available for general use.
PD4	Output – low	O	Available for general use.
PD5	Output – low	O	Available for general use.
PD6	I/O	O	SDQM0
PD7	Output – low	I/O	SDQM1
PE0	Input	O	CS89712 boot-mode selection.
PE1	Input	O	Available for general use. CS89712 boot-mode selection.
PE2	Input	O	

Table 4-7. GPIO Assignments for the CS89712 Development Board

4.5.3 Interrupt Assignments

CS89712 Pin	Signal	Description
nEXTFIQ	-EXTFIQ	Available
nEINT1	-EINT1	Available
nEINT2	-EINT2	Available
EINT3	EINT3	Ethernet Interrupt Request

Table 4-8. Interrupt Assignments for the CS89712 Development Board

For information about internal interrupts, see the *CS89712 Data Sheet*.

4.6 CONTROLLING POWER STATES

4.6.1 *Toggle Switch Settings for Controlling Power / Standby / Battery States*

The switch settings described in Table 4-9 control the operating state for either the battery or an external power source. **Note: toggle down is toggle switched toward the prototype area.**

Switch Label	Signal	Switch Description
BATCHG	nBATCHG	↙ Default: Off ↖ Simulates low battery condition
PWRFL	nPWRFL	↙ Default: Off ↖ Forces CS89712 into Standby state
BATOK	BATOK	↙ Default: No battery installed ↖ Battery OK
EXTPWR	nEXTPWR	↙ External power Off ↖ Default: External power On
Power	Main Power (not a processor pin, other than V_{DD}/V_{PROC}).	↙ Power Off ↖ Power On

Table 4-9. Toggle Switch Assignments

4.6.2 *Setting Toggle Switches to Manage Power*

Switch Name	Signal	Switch Position Signal Active	Switch Position Signal Inactive
BATOK	BATOK	↖	↙
BATCHG	nBATCHG	↖	↙
PWRFL	nPWRFL	↖	↙
EXTPWR	nEXTPWR	↖	↙

Table 4-10. Power Management Switches

4.6.3 Peripheral Power States

Function	Operating	Standby	Idle	nPOR Reset	nURESET
CPU	On	Off	Off	Reset	Reset
PLL/CLKEN	On	Off	On	Off	Off
SDRAM	On	Self Refresh	On	Off	Self Refresh
UARTs	On	Off	On	Reset	Reset
RTC	On	On	On	On	On
TIMERS	On	Off	On	Reset	Reset
LCD	On	Off	On	Reset	Reset
ADC I/F	On	Off	On	Reset	Reset
CODEC I/F	On	Off	On	Reset	Reset
PIC	On	On	On	Reset	Reset
DC – DC	On	Off	On	Reset	Reset

Table 4-11. Power States of the CS89712's Peripherals by Operating State

Chapter 5: Using the Development Board Software

5.1 INTRODUCTION TO THE DEVELOPMENT BOARD SOFTWARE

You can communicate with the CS89712 Development Board and control its features using any of the tools sets listed below. These tool sets are supplied with the kit, along with everything needed to connect them to the CDB89712. Green Hills Multi 2000 runs on a PC, which is connected to the CDB89712 via the PC parallel port through the Wiggler, which is connected to the JTAG connector JP21. The ARM tools run on a PC that is connected to the CDB89712 through a serial cable connected between the COM port on the PC and the Serial Port 1 connector (J9) on the development board. BlueCat Tools run on a PC, which is connected to the CDB89712 through a serial cable connected between the COM port on the PC and the Serial Port 0 (J8) connector on the development board.

Once you set up any of these environments, you have a complete integrated environment to assess the features and functions of the CS89712 and the peripherals designed on the board.

5.1.1 Green Hills Multi 2000

Included in the CD-ROM is a 30-day evaluation copy of Green Hills Multi 2000. This software contains the MULTI 2000 Integrated Development Environment; C/C++ Optimizing Compilers; Instruction Set Simulator; MULTI Multimedia Demonstration (Windows 9x/NT Only); and Documentation in PDF format. The CD-ROM has an excellent “Getting Started Guide” provided by Green Hills. This includes a description of how to get started developing programs using the “hello world” program as an example.

5.1.1.1. Ice_boot

Ice_boot must be loaded into the Flash memory before the Green Hills tools can be installed. The board is shipped with Ice_boot installed. Should it become necessary to reload Ice_boot, use the following procedure:

Power down the board, install a jumper on JP31, power up the board, then press POR (S3).

Open a DOS window, change to the directory “C:\Green\Examples\cirrus_cs89712\ice_boot” and type:

```
download ice_boot.rom
```

Press the Wakeup button (S4).

When the download process is complete, the PC will display the message:

```
Successfully downloaded ice_boot.rom
```

Power down the board, remove JP31, then power up the board. Press POR, wait 2 seconds, then press

Wakeup. The board is now ready to use.

5.1.1.2. Installing the Green Hills tools

To install the Green Hills tools, run Setup.exe from the root directory on the CD-ROM and follow the on-screen directions.

5.1.1.3. Memory map

Table 4-5, “Memory Map for the CS89712 Microcontroller, Booting Ice_boot,” shows the memory map once Ice_boot has been loaded.

5.1.2 ARM Software Development Toolkit

Included in the CD-ROM is a 60-day evaluation copy of the ARM SDT version 2.50. This software contains the Project Manager, C Compiler, Assembler, Linker, Debugger, and ARM instruction set emulator.

There are two applications that are used to build and debug applications:

- ARM Project Manager — The ARM Project Manager is used to develop and build applications for the CS89712 development board.
- ARM Debugger — The ARM Debugger is used to debug applications on the CS89712 development board.

These applications are described fully in the online manuals, but brief instructions for the basic use of these applications is provided in the following sections.

This chapter describes how to use the Project Manager and ARM Debugger to build and debug an application.

5.1.2.1. Installing the ARM SDT

To install the ARM SDT run Setup.exe from the SDT250 directory on the CDROM and follow the on-screen directions.

If you are unfamiliar with the ARM SDT, select the online manuals as one of the components to install. The SDT will have minor problems if it is installed into a directory path that contains a space character; therefore, we recommend that you to **not** install the SDT into the **Program Files** directory.

Caution: The evaluation version of the SDT will create a seemingly useless directory called "c_dilla"; do not remove this directory or its contents or the development version of the SDT will no longer work, will not un-install correctly, and will not be able to be installed again on your system.

5.1.3 LynuxWorks’ BlueCat

A full version of the enhanced BlueCat tools is included in the CDB89712 Development Kit. To install the BlueCat tools refer to the documentation in the BlueCat Tool Kit. Lunuxworks provides 30 days of free BlueCat installation support for the CDB89712. Contact www.lynuxworks.com for details.

5.1.4 Angel Debug Monitor

The example software (See the *Samples* directory on the Development Kit CD-ROM) provided in the development kit is designed to run under control of the Angel debug monitor from ARM. In order to allow the sample code to handle the IRQ interrupt, the version of Angel provided with the development kit (based on Angel 1.2) has been slightly modified to allow the interrupt to be hooked by application code. The Angel software is distributed on sixty CD-ROM, in the sixty directory.

5.1.4.1. Using Angel to Control the Development Board using ARM Tools Debugger

NOTE: (See Greenhill's documentation for connecting to Angel with the Greenhill debugger)

In order to communicate with the development board:

1) Connect one of the supplied NULL modem cables between the Serial Port 0 connector on the development board and any available COM port on the host system.
2) Apply power to the development board.
3) Press the **POR** button on the development board.
4) Wait at least two seconds, then press the **Wakeup** button on the development board.
5) Start the ARM Debugger on the host system.
6) Select **Configure Debugger...** from the **Options** menu.
7) Select **remote_a** as the target environment.
8) Click on the **Configure...** button to select the host COM port to be used. To speed the debugging process, select a baud rate of 115,200. Click on "OK" when done.
9) Click on **OK**. The ARM Debugger should connect to Angel on the development board and then print out a message in the Console Window that is similar to the message found in Figure 5-1.

```
Angel  Debug  Monitor  for  CS89712  Eval  Kit
(Built  with  Serial(x1-2),  no  info)  1.04
(Advanced RISC Machines SDT 2.50)

Angel Debug Monitor rebuilt on Nov 10 1998 at
11:07:40

Serial Rate: 115200
```

Figure 5-1. Successful Connect Message

If there is a problem (such as a bad serial cable, attempting to use the wrong serial port, etc.), the message shown in Figure 5-2 will be displayed:

```
Cannot  open  target:  the  target  is  not
responding.
```

Figure 5-2. Angel Error Message

5.1.4.2. Switching between Development Board Debuggers and the ARMulator when using the ARM SDT

The ARM Debugger can be changed between debugging programs on the CS89712 development board (via Angel) and the ARMulator software ARM emulator at any time. Follow these steps to switch between the two debugging modes:

1) Simply select **Configure Debugger...** from the **Options** menu.
2) Select **remote_a** (for the CS89712 development board) or **ARMulate** for the ARMulator as the target environment, and then click on OK.

Chapter 6: Board Interfaces Reference

6.1 INTRODUCTION

This chapter describes the connectors and headers that allow you to connect the CS89712 Development Board to your host PC and to connect the board to a variety of peripheral devices. It also describes the jumpers and settings that allow you to control the operating mode of these interfaces. Refer to the board layout in Figure 4-3, "Layout of CS89712 Development Board with Major Board Components Identified," for the exact location of the CS89712 Development Board components.

6.2 LOCATION OF MAJOR CONNECTORS DISCUSSED IN THIS CHAPTER

Table 6-1 describes the purpose of the connectors on the CS89712 Development Board and how they are labeled on the board.

Location	Name	Description
J1	POWER	Power Supply
J3	10BASE-T	Ethernet
J8	Serial Port 0	UART0
J9	Serial Port 1	UART1
JP10	JTAG-14	14 pin connector for ICE using JTAG port
JP21	JTAG-20	20 pin connector for ICE using JTAG port

Table 6-1. Description of Connectors

6.3 MAJOR INTERFACES

6.3.1 *Serial Port Interfaces*

The CS89712 is equipped with two UARTs. Both UARTs support data rate up to 115kbps and have 16 byte FIFO on both receive and transmit channels. UART0 also supports the IrDA protocol, which can be enabled under software control.

6.3.1.1. UART0 (J8)

The primary serial port of the CS89712 is UART0. This UART contains many of the standard modem control signals found in a standard UART. The signals not implemented by the CS89712 are DTR, RTS and RI. The development board implements both RTS and RI through GPIO pins. The definition is given in Table 6-2 below.

Serial Port 0 Signal	GPIO Pin
RTS	PB1
RI	PB2

Table 6-2. Serial Port GPIO Implemented Connections

The UART0 signals are converted to and from RS-232 levels by a single device (U13). The DTR signal is driven high or low through jumper JP36. This is done to support serial mice that require all transmitter signals are driven to either a positive or negative voltage. DTR is normally pulled up; it can be pulled down by installing jumper JP36.

6.3.1.2. UART1 (J9)

UART1 is a simplified serial port in the sense that none of the extra control signals are present, only RXD and TXD. This port is driven as with Serial Port 0, using a serial port voltage level shifter. This port has the added feature that the serial port voltage level shifter may be disabled, and the CS89712's driver pins are accessed directly. This is accomplished by installing a jumper on JP35, and using JP34 to access the port pins. JP34 is outlined in Table 6-3 below.

Pin #	Name
1	TXD2
2	RXD2
3	Ground

Table 6-3. Serial Port Logic Level Header (JP34)

6.3.1.3. Controlling Serial Port Communications

Table 6-4 describes the jumper setting for enabling / disabling COM1 and COM2 communications and for setting the COM1 DTR signal high or low.

Jumper	Name	Description	
JP35	COM2 ENABLE / DISABLE	● ● ○ ○	Enabled. (Factory setting) Disabled. Notice that this only affects the driver IC, and will not disable the port in the CS89712 itself. The intent of this jumper is to permit use of logic level RS232 communications via JP34.
JP36	COM1 DTR HIGH / LOW	● ● ○ ○	DTR set high. This should be used for most applications. (Factory setting) DTR set low. Used to disable communications, or when DTR is negative true.

Table 6-4. Serial Port Control Jumper Assignments

6.3.2 Infrared (IrDA) Interface

The development boards IrDA port supports Infrared communications for such applications as data transfer between hand-held devices and printing files on infrared-enabled printers. The IrDA port supports data transfers of up to 115.2 kbps. The IrDA port on the CS89712 is shared with UART1. The processor can be configured to use either the standard serial port pins, or the LEDDRV and PHDIN pins to handle infrared signals. See the CS89712 data sheet for a full explanation of this configuration.

6.3.3 Twisted Pair Ethernet Interface (J3)

The Ethernet controller is a 10Base-T controller. A detailed discussion of the Ethernet port is beyond the scope of this document. Developers wishing to know more should refer to the CS89712 data sheet.

6.3.4 Headers

Table 6-5 describes the headers for the CS89712 Development Board.

Location	Name	Description
JP21	MULTI-ICE	Connection to Multi-ICE or Wiggler
JP24	MULTI-ICE	Connection to Multi-ICE.
JP34	TTL SERIAL PORT 1	TTL connection to Serial Port 1
JP19, JP20, JP22, JP23	Expansion headers for user defined options	Refer to schematic for details on header pin assignments.
JP15	RS232 EN-DIS	Enable / disable Serial Port 1
JP31	TTL/SER2	TTL I/O for Serial Port 1
JP27-30	CPU CONFIG	Configures CPU clock speed and boot operation.
J4-7	PROCESSOR PINS	Intended for future product development. In future, this board may be manufactured with the processor depopulated and a separate processor carrier board dropped onto these sites.

Table 6-5. Header Assignments

6.3.5 Expansion Interface

This expansion interface is intended to support the addition of small add-on boards. The interface exposes sufficient signals to allow byte-wide interfacing to peripheral devices. It should be noted that the interface is 3.3V only.

For a detailed explanation of this interface refer to the *CS89712 Data Sheet*.

Description	Pin No.	Pin No.	Description
D0	1	2	V_{DD} (3.3V)
D1	3	4	A0
D2	5	6	A1
D3	7	8	A2
D4	9	10	A3
D5	11	12	nMWE (Write Enable)
D6	13	14	nEXTFIQ (Interrupt)
D7	15	16	PD1 (GPIO)
nCS5 (Chip Select)	17	18	GND
nMOE (Output Enable)	19	20	GND

Table 6-6. Expansion Interface

6.3.6 Data Input and Control Interfaces

6.3.6.1. Diagnostic LED

A single LED may be driven by software as a visual indicator of activity. The GPIO bit PD0 drives the LED. On a power-on reset the LED is not lit. When the PD0 bit is set to an output driving the port bit high will light the LED.

Chapter 7: Board Specifications

7.1 INTRODUCTION

This chapter describes the specifications for the development board and the components on the board. It also describes the optimum operating and storage environment that ensures that the development board will operate at the performance levels described in the *CS89712 Data Sheet*.

7.2 PHYSICAL SPECIFICATION

The development board has the physical characteristics described in the following sections.

7.2.1 Printed Circuit Board (PCB)

7.2.1.1. PCB Form Factor

The PCB form factor is 19.6 cm x 20.2 cm (7.7” x 8”). A picture of the board is shown in Figure 4-3, “Layout of CS89712 Development Board with Major Board Components Identified,” .

7.2.1.2. PCB Construction Materials

The PCB is constructed from materials with a flame rating of 94V0. This rating meets the self-extinguishing characteristics required by safety agencies in countries of sale.

7.2.1.3. Connector Characteristics

Processor Daughter Module Headers (J4-J7)

The four connectors intended for use as daughter module headers are arranged around the processor and are intended for future processor development. Example part: Samtec - TSW-125-07-G-D.

Expansion Header (J22)

These headers bring out the pins for the keyboard, LCD, and other interfaces. The headers are 0.100” spacing, 0.025” square posts. Example part: Samtec TSW-120-07-G-S.

Ethernet (J3)

The Ethernet connector used in this design is a standard right angle 10Base-T RJ45. Example part: CorCom - RJ45-8L2-S.

Serial Ports (J8, J9)

The serial port connectors are male, right angle standard density 9-contact DSUBs. Example part:

AMP - 787203-1.

Power (J1)

DC Power Jack, 2.5mm Mini power connector, 2 position 0.1 pin. Example part: Switchcraft - RAPC712.

7.3 RECOMMENDED OPERATING AND STORAGE ENVIRONMENTS

7.3.1 Temperature

The CS89712 Development Board can be stored safely at temperatures ranging from -20 to 85°C. The CS89712 Development Board functions reliably in an ambient operating temperature of 0-70°C, inclusive.

7.3.2 Humidity

The CS89712 Development Board operates reliably in storage and/or operating in relative humidity of 10% to 95% (non-condensing) in the appropriate temperature ranges specified above.

7.3.3 Air Flow

The CS89712 Development Board operates reliably in storage and/or operating in static air at temperatures and relative humidity specified above.

Appendix A: Development Kit Software Reference

A.1 INTRODUCTION

The CS89712 Development Kit includes the library of software routines and sample code described in this chapter. These routines are distributed on the Green Hills CD-ROM. After installation, the source code, include files, build files and documentation for these routines are located in the *Green\Examples\cirrus_cs89712* directory of the target drive.

A.1.1 bnchtest.c

This example demonstrates the EEPROM and Ethernet port testing procedure. This application includes a loopback test for Ethernet port and an EEPROM test in which we can write and read formatted data to the eeprom.

A.1.2 CustomRunTimeErrorChecking.c

The Green Hills C runtime libraries offer the option of run-time error checking of common (but potentially nasty) programming bugs such as null pointers, memory allocation errors, array boundary overflows and others.

Normally these runtime error checks are reported in the debugger: execution will halt and an error message will be displayed in the debugger's command pane.

Certain applications may require these runtime error tests to be performed during in-field testing, without a debugger connection.

This example shows how to create a custom runtime error checking handler function for this purpose. If a runtime error condition is detected, the user can supply their own function to handle and report the condition.

A.1.3 helloworld.c

Demonstrates a simple "hello world" application using the I/O window as stdio.

A.1.4 ledblink.c

This example demonstrates how to control a led by writing the relative register. The led will blink constantly after running.

A.1.5 MemoryChecking.c

Applications which use dynamic memory management (malloc / free) must ensure that all malloc'd memory blocks are freed after use.

If allocated memory is not freed, then that memory is no longer available for re-allocation to the program, thus reducing the total available memory. This condition is known as a memory leak, and can cause an application to fail because of a lack of available memory.

A.1.6 rom_ledblink.c

This project is an example that shows how to create a stand-alone application. It creates the file "rom_ledblink.bin" which is a binary image which should be flashed to the board. This can be done by following the next steps:

1) Connect a NULL modem cable between com1 of your host machine and UART0 of the CS89712.

2) With no power to the board connect jumper31.

3) Run the download.exe program with the following options:

"download rom_ledblink.bin"

4) Power the board and press the wakeup button.

5) Once the download is complete LED D7 will light up.

6) Power off the board.

7) Remove jumper31.

8) Power the board and press the reset button.

9) Wait 2 seconds and then press the Wakeup button.

LED D7 will begin to blink at a constant rate.

A.1.7 RunTimeErrorChecking.c

Simple programing errors which cannot be detected at compile-time can be very difficult to debug. Errors such as indexing outside of an array boundary can overwrite other variable giving erroneous values later in your application.

MULTI's run-time error checking will halt execution, warning that a run-time error has occurred during your debug session.

If it is undesirable for your application to halt at a run-time error, see the "CustomRunTimeErrorChecking" example.

A.1.8 SortByName.c

This example demonstrates a wide variety of MULTI's capabilities.

The application is a straightforward sorting demonstration: it will open a file on the host system, read in a number of records from that file, sort them, and print out the sorted list in the I/O window.

- Host I/O Capability:

MULTI provides an interface by which an embedded application can access the host machine's disk drives, keyboard, and console (via the INPUT/OUTPUT window).

- Runtime Error Checking:

There are a number of intentional errors in this demonstration program. These errors will not cause the application to fail, and are composed of completely legal C statements. Program errors such as these are easy to create, can manifest themselves unpredictably, and are difficult to reproduce and debug. MULTI's runtime error checking libraries will detect these errors at the first execution, allowing you to concentrate on more important issues.

- Mixed-Language Debugging Capability:

This application is composed of C and C++ source files. MULTI can seamlessly build and debug mixed-language applications written in C, C++, EC++, FORTRAN, or Assembly.

- Formatted Display of Variables / Structs / Arrays:

This demonstration uses a linked list of structs, arrays of pointers, bitfields and other complex variables. MULTI provides exceptional formatted visibility into these program elements.

• Notes •

SMART Analog™