GLOMATION

Embedded Single Board Computer

GESBC-9312-sx

User's Manual

Table of Contents

List of Tables

Chapter 1 – Introducing the GESBC-9312 Single Board Computer

GESBC-9312 Overview

The GESBC-9312 a feature rich compact sized single board computer based on Cirrus Logic EP9312 processor. With a large peripheral set targeted for a variety of applications, the GESBC-9312 is well suited for:

- Thin Client Computers for Business and Home
- Internet Access Devices
- Industrial Computers
- Point of Sale Terminals
- Medical Instrument
- Test and Measurement Equipment

Advanced Features

The heart of the GESBC-9312 is the EP9312 which is the one in a series of ARM920T-based processors. The ARM920T microprocessor core with separate 16 Kbyte 64-way set-associative instruction and data caches is augmented by the MaverickCrunch™ co-processor. This enables faster than real-time compression of audio CDs. The proprietary MaverickKey™ unique hardware programmed IDs provide an excellent solution to the growing concern over secure Web content and commerce. MaverickKey IDs can also be used by OEMs and design houses to protect against design piracy by presetting ranges for unique IDs.

The EP9312 is a high-performance, low-power RISC-based device built around a single ARM920T microprocessor core. The ARM920T on the EP9312 functions with a maximum operating clock rate of 200MHz and a power usage between 100mW and 750mW (dependent upon clock speed). The ARM core operates from a 1.8V supply while the I/O operates at 3.3V. The low power consumption makes it an idea platform for battery operated applications.

A high performance 1/10/100 Mbps Ethernet Media Access Controller (EMAC) is included along with external interfaces to SPI and I2S Audio, Raster/LCD (with touch screen interface), IDE storage peripherals, and keyboard or keypad peripherals. A three-port USB host and three UARTs are included as well. The list below summarizes the advanced features of the GESBC-9312.

- 200MHz Processor Core – ARM920T with MMU
- MaverickCrunch™ Math Engine
- MaverickKey™ Security
- 64M SDRAM, 8 - 32M FLASH
- IDE Interface

- Ethernet Media Access Controller (EMAC)
- Raster / LCD Interface
- On board VGA with optional TV-out
- Touch Screen Interface with 12-bit Analog-to-Digital Converter (ADC), optional additional 8 Channel 12 bit ADC with 4 bit Digital I/O
- Optional 4 Channel Digital-to-Analog Converter
- 64-Keypad Interface or General Purpose Input/Output
- Universal Asynchronous Receiver / Transmitters (UARTs)
- Triple Port USB Host
- Real-Time Clock
- Hardware Debug Interface

Figure 1 below show a picture of the GESBC-9312 Single Board Computer, while Figure 2 provides a block diagram of typical input and output device interface connections on the GESBC-9312.

Figure 1 GESBC-9312 Single Board Computer

EP9312

The GESBC-9312 is shipped with the Cirrus Logic EP9312 processor. For more information regarding the EP9312 processor please see the EP9312 datasheet.

SDRAM

The GESBC-9312 is shipped with 64MBytes of SDRAM.

FLASH

The GESBC-9312 is shipped with 8MBytes of asynchronous Intel Strata-Flash, can be upgraded to 32MB.

IDE Interface

The IDE interface on the GESBC-9312 provides a standard connection to ATAPI compliant devices.

Real Time Clock

The GESBC-9312 has on-chip real time clock device with software trim.

Video Interface (Raster / LCD interface)

The Raster / LCD interface provides data and interface signals for a variety of display types.

- LCD Interface
    - The GESBC-9312 is shipped LCD display interface
- Touch Screen
    - The GESBC-9312 is shipped with a 4-wire, touch screen interface
- VGA
    - The GESBC-9312 is shipped with an analog VGA connection
- Video Encoder
    - The GESBC-9312 provides a composite video and S-Video output connections

Keypad

The GESBC-9312 supports an 8x8 keypad array.

USB

The GESBC-9312 is shipped with three type A USB host connections.

UART 1

The GESBC-9312 is shipped with a 9-pin modem interface comprised of the following signals:

- Transmit
- Receive
- Clear to Send
- Data Set Ready
- Data Terminal Ready
- Ready to Send
- Ring Indicator

UART 2

The GESBC-9312 is shipped with 3 wire UART interface on UART 2.

UART 3

The GESBC-9312 is shipped with 3 wire UART interface on UART 3.

Audio Input

The GESBC-9312 is shipped with a single stereo audio input.

Audio Output

The GESBC-9312 is shipped with a single stereo audio output.

AC-97

The GESBC-9312 is shipped with a Cirrus Logic CS4202 AC-97 codec. (Please note the use of the CS4202 or CS4228 is mutually exclusive.)

Ethernet

The GESBC-9312 is shipped with a complete physical and MAC subsystem that is compliant with the ISO/TEC 802.3 topology for a single shared medium with several stations. The EP9312 supports 1/10/100 Mbps transfer rates and interfaces to industry standard physical layer devices such as the Cirrus Logic CS8952.

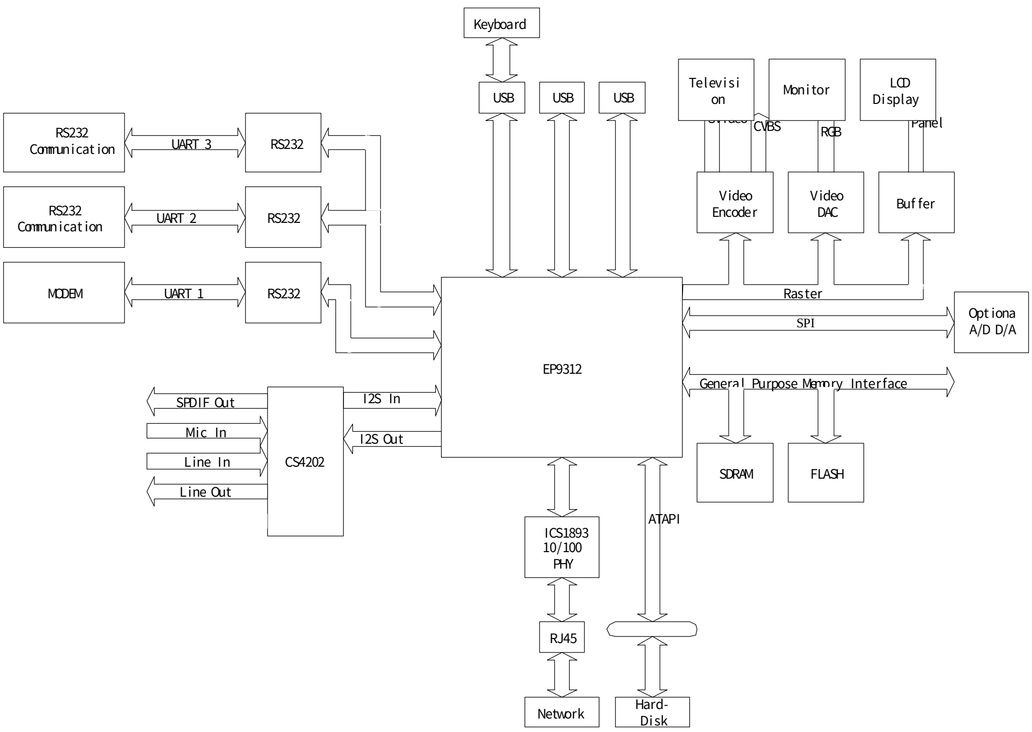


Figure 2 Typical Input / Output Device Interface Connections On The GESBC-9312.

Chapter 2 – Getting Started

This chapter describes the GESBC-9312 working environment and familiarizes the user with its components and functionality. This chapter contains the following sections:

- Assembly and Connections
  - Describes how to assemble and connect components to the GESBC-9312 Single Board Computer

- Operation
  - Describes how to operate the GESBC-9312 Single Board Computer

Assembly and Connections

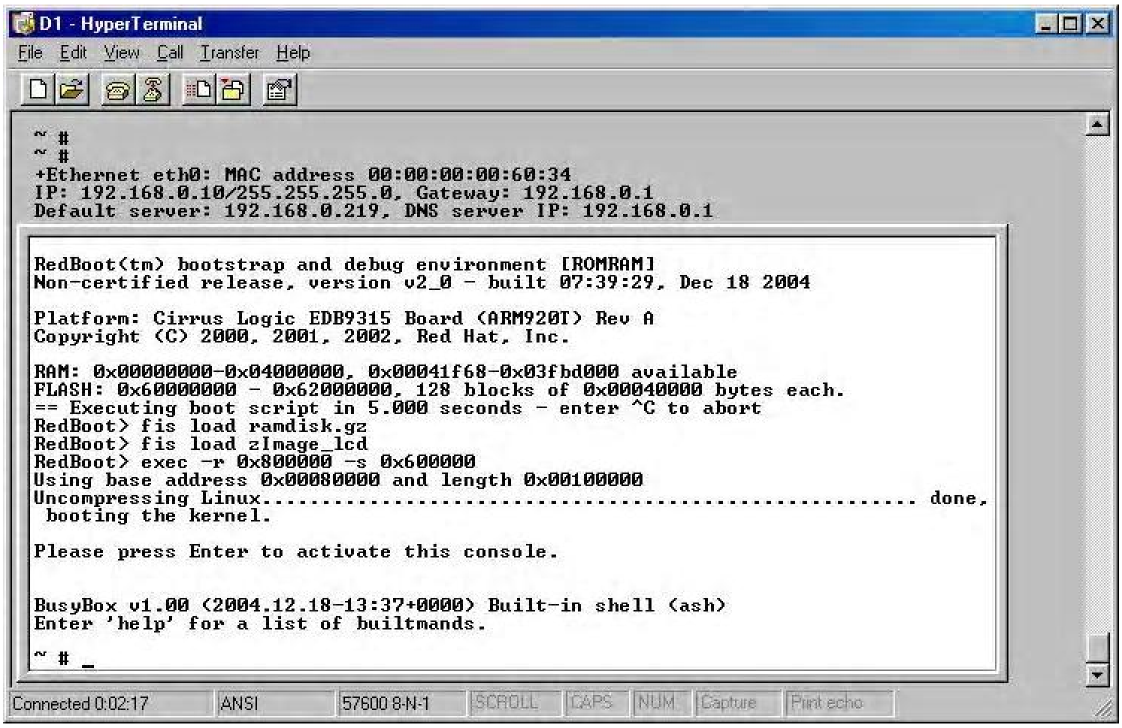
In order to use the GESBC-9312 the user must first assemble and connect the peripherals to the GESBC-9312, as described in the following procedure.

1. Place the GESBC-9312 on a static free surface.
2. Make sure all of the jumpers are in the factory default position. The unit is shipped in a factory default configuration. If the user is uncertain that the GESBC-9312 has the jumpers in the factory default configuration, please see Chapter 4 – Linux Software Description.
3. Connect power to the board.
4. Connect null modem serial cable between GESBC-9312 UART 1 and PC/terminal serial port.
5. Launch a terminal emulator, such as HyperTerminal, or minicom, on the PC configured to connect to the serial port of the GESBC-9312. Configure the serial port with the following parameters: 57600 bits per second, 8 data bits, no parity, 1 stop bit, no flow control.
6. Connect the board to a local area network (optional)

Operation

The startup procedure for the GESBC-9312 is straightforward. First, the connection of the power harness is required. Second, the null modem serial interface cable must be connected to the UART1 connector. Third, connect the GESBC-9312 to a network that has Internet access. It is recommended that all other cables be tested to determine they are properly seated.

A few seconds after applying power to the GESBC-9312, debug information will be displayed on the terminal program. The following figure shows what this should look like.



Please see Chapter 4 – Linux Software Description for more detail regarding of the software functionality.

Configurations

Jumpers are used to configure the GESBC-9312 to operate in different mode. The following table lists all the settings for each jumper.

Table 1 System configuration

Jumper	Description
JP9	Boot mode: 1 and 2 - serial boot 2 and 3 - flash boot
JP11	Boot mode: 1 and 2 – internal ROM 2 and 3 – external memory

Chapter 3 – GESBC-9312 Function Blocks

This chapter provides detailed description of each functional block of GESBC-9312 single board computer.

EP9312

The GESBC-9312 Single Board Computer uses the Cirrus Logic EP9312 as the core processor on this development board. The top-level features of EP9312 processor are the following:

- ARM920T RISC Core Processor
- 200 MHz / 200 MIPS Performance
- 16 Kbyte Instruction Cache
- 16 Kbyte Data Cache
- Linux and Windows CE enabled MMU
    - Note: Cirrus Logic to supply either a Linux port or a Windows CE port, including the respective board support package (BSP).
- 100 MHz System Bus
- MaverickCrunch™ Math Engine
- MaverickKey™ Security Features
- 32 bit SDRAM Interface (Up To 4 Banks)
- 32 / 16 bit SRAM / FLASH / ROM Interface
- Serial EEPROM Interface
- EIDE Interface
- 10 / 100 Mbps Ethernet MAC
- Three UARTs
- Three-port USB Host
- LCD / Raster Interface With Graphics Accelerator
- Touch Screen Interface With ADC
- 8 x 8 Keyboard Scanner
- SPI Port
- Serial Audio Interface
- JTAG Interface

More detailed information regarding the EP9312 processor can be found at www.cirrus.com and on the enclosed disk.

SDRAM

The EP9312 features a unified memory address model where all memory devices are accessed over a common address and data bus. The EP9312 can support a minimum of 1 to a maximum of 4 banks of 32-bit 66 or 100 MHz SDRAM. Additionally, the GESBC-9312 supports 64 Mbyte SDRAM density. The 64 Mbyte option is supported by using two 256 Mbit SDRAMs in a 4M x 16 x 4 bank architecture. The K4S56132 SDRAM manufactured by Samsung is a part that matches this requirement. The features of the Samsung SDRAM include the following:

- JEDEC Standard 3.3V Power Supply
- LVTTL Compatible with Multiplexed Address
- 4 Bank Operation
- MRS Cycle with Address Key Programs
- CAS Latency of 2 Or 3
- Burst Length of 1, 2, 4, 8, and Full Page
- Burst Type of Sequential and Interleave
- All Inputs Are Sampled on Rising Edge of Clock
- DQM for Masking
- Auto and Self Refresh
- 64ms Refresh Period

FLASH

As previously stated, the EP9312 features a unified memory address architecture. The EP9312 can support either NAND or NOR types of non-volatile flash memory for program code storage. The GESBC-9312 is shipped with 8 Mbytes of flash memory, which is provided by two Intel E28FXXXJ3A Strata FLASH memory devices. The Intel Strata Flash memory device features include the following:

- 150ns Initial Access Speed
- 25 ns Asynchronous Page-Mode Reads
- 32-Byte Write Buffer
- Program and Erase Suspend Support
- Flash Data Integrator, Common Flash Interface Compatible
- 128-bit Protection Register
- 64-Bit Unique Device Identifier
- 64-Bit User Programmable OTP Cells
- Absolute Protection with VPEN Tied to Ground
- Individual Block Locking
- High Density 128 Kbyte Blocks

IDE Interface

The GESBC-9312 Single Board Computer provides one IDE interface. The IDE interface will allow a standard connection to an external, ATAPI compliant device that can be connected to off the shelf CD-ROM, DVD-ROM and hard disk drives. The IDE interface provided on the GESBC-9312 supports the following operating modes:

- PIO Mode 4
- Multi-Word DMA Mode 2
- Ultra DMA Mode 2

Table 2 J21 IDE Connector

Pin Number	Signal Name	Pin Number	Signal Name
1	RST	2	GND
3	D7	4	D8
5	D6	6	D9
7	D5	8	D10
9	D4	10	D11

11	D3	12	D12
13	D2	14	D13
15	D1	16	D14
17	D0	18	D15
19	GND	20	KEY
21	DMARQ	22	GND
23	IOWR	24	GND
25	IORD	26	GND
27	IORDY	28	CSEL
29	DAMACK	30	GND
31	IRQ	32	IOCS16
33	A1	34	PDIAG
35	A0	36	A2
37	CS0	38	CS1
39	DASP	40	GND

Video Interface (Raster / LCD interface)

The Video Interface, also known as the Raster/LCD interface, on the EP9312 provides data and interface signals for a variety of display types. This interface provides programmable video interface timing for non-interlaced flat panel or dual scan displays. Resolutions up to 1280 x 1024 are supported from a unified SDRAM based frame buffer. Features specific to LCD display support are the following:

- Provides Timing and Interface Signals For Digital LCD / TFT displays
- Fully Programmable:
    - Non-interlaced or Dual Scan
    - Color or Grayscale Displays
- Dedicated Data Path to SDRAM Controller for Improved System Performance
- Pixel Depths of 4, 8, Or 16-Bits per Pixel
    - 256 Levels of Grayscale Support
- Hardware Cursor up to 64 x 64 x 2 Pixels
- 256 x 18 Color Lookup Table
- Hardware Blinking
- 8-bit Interface for Low-End Displays

The LCD interface and touch screen input are brought out by a single 2x25 pin header. The signal designation is shown in the following table.

Table 3 J6 LCD Connector

Pin Number	Signal Name	Pin Number	Signal Name
1	3.3V	2	5V
3	3.3V	4	5V
5	GND	6	GND
7	DCLK	8	BLANK
9	HSYNC	10	VSYNC
11	GND	12	GND
13	RED0	14	RED1
15	RED2	16	RED3
17	RED4	18	RED5
19	GRN0	20	GRN1
21	GRN2	22	GRN3
23	GRN4	24	GRN5
25	BLU0	26	BLU1
27	BLU2	28	BLU3
29	BLU4	30	BLU5
31	NC	32	NC
33	GND	34	GND
35	12V	36	12V
37	/SLEEP	38	BRIGHT
39	SET1	40	SET2

VGA

The GESBC-9312 Single Board Computer is shipped with support for an analog VGA connection. The digital video data is converted into an analog form by using a triple D/A converter, such as the ADV7123 from Analog Devices. This device provides the following features:

- > 80 MSPS Throughput Rate
- Triple 10-Bit D/A Converters

The VGA interface is brought out through a female HDSUB-15 connector. The signal designation is listed in the table below.

Table 4 J15 VGA connector

Pin Number	Signal Name	Pin Number	Signal Name
1	RED	2	GREEN
3	BLUE	4	ID2
5	GND	6	GND

7	GND	8	GND
9	NC	10	GND
11	ID1	12	ID0
13	HSYNC	14	VSYNC
15	NC		

Video Encoder

The GESBC-9312 Single Board Computer is shipped with support for a composite video output as well as an S-Video output. The digital data is converted into analog composite and S-Video formats the video encoder, such as the CX25871 from Conexant. This video encoder supports both NTSC and PAL formats.

The S-Video interface is brought out through a single 1x4 2.54mm header. The signal designation is listed in the following table.

Table 5 J18 S-Video connector

Pin Number	Signal Name	Pin Number	Signal Name
1	LUMIN	2	CHROM
3	GND	4	GND

Table 6 J17 Composite video connector

Pin Number	Signal Name	Pin Number	Signal Name
1	Video	2	GND

Touch Screen

The GESBC-9312 Single Board Computer is shipped with support for a touch screen panel. The EP9312 processor performs the sampling, averaging, ADC range checking, and control for a wide variety of analog, resistive touch screens. The GESBC-9312 will support either a 4 or 8-wire touch screen interface. The touch screen interface is brought out through a single 2x5 2.54mm header. The signal designation is listed in the following table.

Table 7 J9 Touch screen input / On-chip A-D Converter

Pin Number	Signal Name	Pin Number	Signal Name
1	sXp	2	Xp
3	Xm	4	sXm
5	Ym	6	sYm
7	sYp	8	Yp
9	GND	10	GND

USB

The GESBC-9312 Single Board Computer provides three type A USB host connections. The EP9312 USB host controller is configured for three root hub ports and features an integrated transceiver for each port. The EP9312 integrates three USB 2.0 Full Speed host ports. These ports are fully compliant to the OHCI USB 2.0 Full Speed specification (12 Mbps). The controller complies with the OHCI specification for USB Revision 1.1. The USB ports are brought out by a 2x5 2.54mm header and a 1x5 2.54mm header. The following tables list the signal designation.

Table 8 J13 USB 1 and USB 2 connector

Pin Number	Signal Name	Pin Number	Signal Name
1	VDD	2	VDD
3	USBM0	4	USBM1
5	USBP0	6	USBp1
7	GND	8	GND
9	NC	10	NC

Table 9 J2 USB 3 connector

Pin Number	Signal Name	Pin Number	Signal Name
1	VDD	2	USBM2
3	USBP2	4	GND
5	NC		

UART 1

The GESBC-9312 Single Board Computer is shipped with support for a 9-pin UART interface. The UART interface is provided via a 2x5 2.54mm header. The signal designation is listed in the following table.

Table 10 P1 UART1 connector

Pin Number	Signal Name	Pin Number	Signal Name
1	DC	2	RX
3	TX	4	DTR
5	GND	6	DSR
7	RTS	8	CTS
9	RI	10	NC

UART 2

The GESBC-9312 Single Board Computer is shipped second UART interface. The UART interface is provided via a 1x4 2.54mm header. The signal designation is listed in the following table.

Table 11 P3 UART2 connector

Pin Number	Signal Name	Pin Number	Signal Name

1	NC	2	RX
3	TX	4	NC
5	GND	6	NC
7	NC	8	NC
9	NC	10	NC

UART 3

The GESBC-9312 Single Board Computer is shipped with third UART interface. This interface is provided via a 1x4 2.54mm header. The signal designation is listed in the following table.

Table 12 P2 UART3 connector

Pin Number	Signal Name	Pin Number	Signal Name
1	NC	2	RX
3	TX	4	NC
5	GND	6	NC
7	NC	8	NC
9	NC	10	NC

EGPIO

GESBC-9312-sx provides EGPIO signals that are not used by the system through J10. The signal designation is listed in the following table. Care must be taken when changing the value of port registers that controls EGPIO since other pins are used by the system are controlled by the same set of registers.

Table 13 J10 EGPIO

Pin Number	Signal Name	Pin Number	Signal Name
1	EGPIO3	2	EGPIO4
3	EGPIO5	4	EGPIO6
5	EGPIO7	6	EGPIO8
7	EGPIO9	8	EGPIO10
9	EGPIO11	10	EGPIO12
11	EGPIO13	12	EGPIO14
13	GND	14	GND
15	GND	16	GND

AC-97 and Audio Input/Output

The GESBC-9312 Single Board Computer is shipped with support for an AC-97 audio codec interface. The Cirrus Logic CS4202 is used to implement this functionality and interface. The GESBC-9312 Single Board Computer is shipped with support for a single stereo audio input, single stereo audio output and mono microphone input. The audio singles are provided via 3 1x3 2.54mm headers. The signal designation is listed in the following table.

Table 14 J4 Audio Output

Pin Number	Signal Name
1	Line out L
2	Line out R
3	GND

Table 15 J5 Audio Intput

Pin Number	Signal Name
1	Line in L
2	Line in R
3	GND

Table 16 J7 Mic Input

Pin Number	Signal Name
1	Mic in
2	GND
3	NC

Ethernet

The GESBC-9312 Single Board Computer is shipped with support for a complete Ethernet interface. The EP9312 contains a MAC subsystem that is compliant with the ISO/TEC 802.3 topology for a single shared medium with several stations. The Media Access Controller (MAC) within the EP9312 supports 1/10/100 Mbps transfer rates and interfaces to industry standard physical layer devices. The GESBC-9312 is shipped with the ICS1983 100Base-X / 10Base-T Transceiver device which provides the physical layer interface. The standard RJ-45 connector, J1, is used for network connection.

SPI bus

The GESBC-9312-sx Single Board Computer is shipped with a SPI expansion bus header for peripheral expansion. The signal designation is listed in the following table.

Table 17 J19 SPI connector

Pin Number	Signal Name

1	SFRM
2	SSPRX
3	GND
4	SCLK
5	GND
6	SSPTX

Keypad/Digital-IO

The GESBC-9312 Single Board Computer is shipped with support for 8x8 keypad. It can be also used as digital I/O. Each individual pin can be programmed as input or output. The keypad/digital-IO interface is provided via a 2x10 2.54mm header. The signal designation is listed in the following table.

Table 18 J8 Keypad/Digital-IO connector

Pin Number	Signal Name	Pin Number	Signal Name
1	ROW0	2	ROW1
3	ROW2	4	ROW3
5	ROW4	6	ROW5
7	ROW6	8	ROW7
9	COL0	10	COL1
11	COL2	12	COL3
13	COL4	14	COL5
15	COL6	16	COL7
17	GND	18	GND
19	GND	20	GND

JTAG

The GESBC-9312 Single Board Computer is shipped with a 20 pin JTAG connector. The JTAG provides the user with the ability to program CPLD in circuit and debug system level programs. The signal designation is listed in the following table.

Table 19 JP6 JTAG connector

Pin Number	Signal Name	Pin Number	Signal Name
1	VDD3.3	2	VDD3.3
3	TRST	4	GND
5	TDI	6	GND
7	TMS	8	GND
9	TCK	10	GND
11	TCK	12	GND

13	TDO	14	GND
15	RESET	16	GND
17	NC	18	GND
19	NC	20	GND

A/D and D/A

The GESBC-9312 Single Board Computer provides support for optional 12 bit 8 channel A/D and 4 channel 12 bit D/A. The A/D is provided by TI ADS7870 which is a 12 bit 8 channel analog to digital converter with programmable gain amplifier. It also provide 4 programmable digital I/O. The maximum sampling rate of ADS7870 is 100 ksps. The 8 single ended analog inputs can be also configured as 4 pairs of differential input channels. The optional D/A is provided via TI DAC7554 which is a voltage output 12 bit 4 channel digital to analog converter. The A/D and D/A interface is provided via a 2x10 2.54mm header. The signal designation is listed in the following table.

Table 20 J30 A/D and D/A connector

Pin Number	Signal Name	Pin Number	Signal Name
1	DIO0	2	DIO1
3	DIO2	4	DIO3
5	VDD5	6	DGND
7	AIN0	8	AIN4
9	AIN1	10	AIN5
11	AIN2	12	AIN6
13	AIN3	14	AIN7
15	AGND	16	AGND
17	AOUT0	18	AOUT1
19	AOUT2	20	AOUT3

Reset switch

A reset switch can be connected to S1 to manually reset the system.

Table 21 S Reset switch

Pin Number	Signal Name
1	Reset
2	GND

Power Requirement

The GESBC-9312 Single Board Computer requires regulated 5v and 12v DC. If the LCD operation is not needed, the 12v DC input can be left open. The board can function well with a single 5v DC supply. The input connector signal designation is listed in the table below.

Table 22 J22 Power supply connector

Pin Number	Signal Name
1	5V DC
2	GND
3	GND
4	12V DC

Chapter 4 – Linux Software Description

Overview

This chapter provides information regarding the Linux software that is shipped with the GESBC-9312 Board. The software included with the board is Linux with a few test applications and network utilities. The Linux software provides the user with the ability to test some of the subsystems on the GESBC-9312 board. The download utility provides a means to program a binary image into the flash memory on the GESBC-9312.

GESBC-9312 Linux Code

The pre-programmed software provides the user with the opportunity to test some of the subsystems on the GESBC-9312 via Linux. This software is programmed into the system FLASH located on the board prior to shipment. The binary image of the shipped code is included on the CD that ships with the board.

Raster Interface

Functionality of the raster interface is demonstrated by the QT/Embed user interface. The user interface is configured for 640 x 480, 16 Bit color depth

Serial Port Interface

The functionality of the serial interface can be demonstrated by looking at the debug messages while the system boot-ups and operates. The setting for the serial interface is described in chapter 2.

Touchscreen Interface

The functionality of the touchscreen can be supported by user supplied hardware.

USB Interface

The functionality of the USB interface can be shown by hooking up a user supplied USB device.

Audio Output Interface

Audio out functionality can be shown by attaching powered speakers (not included) and turning on the power to the GESBC-9312. Running the included MP3 player program and the sound should be heard from the speaker.

Ethernet Interface

Ethernet is automatically detected by the Linux kernel and a DHCP client will try to lease network address from connected DHCP server.

Download Utility

The download utility provides the user with a tool for programming the flash memory on the GESBC-9312 with a binary image. The following procedure will allow in-circuit programming of the flash memory via the EP9312 processor:

1) Power the board off.
2) Connect null-modem serial cable to UART1.
3) Set JP9 to connect pin 1 and 2 (JP9 factory default is 2 and 3).
4) Stop any program that might use the serial port that is connected to GESBC-9312.
5) Run download utility (assuming download_win.exe located in same directory as binary image)
   download_win binary_image_filename.bin
6) "Waiting for board to wake up…" message is displayed.
7) Power the board on.
8) Messages displayed regarding erasing, then programming the flash.
9) "Successfully programmed binary_image_filename.bin" message displayed upon programming completion.
10) Power the board off.
11) Set jumper JP9 to 2 and 3.
12) Power the board on.

Redboot

RedBoot provides a simple interface for loading operating systems and applications onto the GESBC-9312-sx board. It can also serve as a debug platform for standalone programs using the GDB stub that is built into RedBoot.
RedBoot uses a serial console for its input and output. The default serial port setting is 57600,8,N,1. It also supports the built-in Ethernet port and a flash file system and general flash programming.

The board is shipped with Redboot pre-installed. Please refer to Download Utility section for instructions to reload Redboot. Please refer to documents at ECOS website http://ecos.sourceware.org regarding how to rebuild Redboot.

Loading Linux Kernel and root File System

The Redboot boot-loader provides two ways to load Linux kernel and file system into FLASH memory, by Ethernet network and TFTP server or serial connection. The network connection method provides fast loading but if not available, serial port connection method can be used.

After power on the GESBC-9312 board, the following message should be shown on the terminal console on the host PC connected to the GESBC-9312-sx board[1].

```
+FLASH configuration checksum error or invalid key
EP93xx - no EEPROM, static ESA, or RedBoot config option.
No network interfaces found

RedBoot(tm) bootstrap and debug environment [ROMRAM]
Non-certified release, version v2_0 - built 07:39:29, Dec
18 2004

Platform: Cirrus Logic EDB9315 Board (ARM920T) Rev A
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x00000000-0x04000000, 0x00041f68-0x03fbd000 available
FLASH: 0x60000000 - 0x60800000, 32 blocks of 0x00040000
bytes each.
RedBoot>
```

It's possible to use bootp of Redboot to acquire network address automatically. For situation it is not available, the following procedure can be used to configure a static IP address for the SBC.

```
RedBoot> fconfig
Run script at boot: true
Boot script:
Enter script, terminate with empty line
>> fis load ramdisk
>> fis load zImage
>> exec -r 0x800000 -s 0x600000
>>
Boot script timeout (1000ms resolution): 1
Use BOOTP for network configuration: false
Gateway IP address:
Local IP address: 192.168.0.112
Local IP address mask:
Default server IP address:
DNS server IP address:
Set eth0 network hardware address [MAC]: true
eth0 network hardware address [MAC]: 0x00:0x00:0x00:0x00:0x80:0x21
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
Update RedBoot non-volatile configuration - continue (y/n)? y
... Erase from 0x60780000-0x60781000: .
... Program from 0x03fbe000-0x03fbf000 at 0x60780000: .
RedBoot>
```

[1] A slightly different message will be displayed if the FLASH memory has been initialized and programmed before.

The Redboot FLASH file system must be initialized in order to store data in the FLASH file system. The following procedure is used to initialize the Redboot FIS.

```
RedBoot> fis init
About to initialize [format] FLASH image system - continue (y/n)? y
*** Initialize FLASH Image System
    Warning: device contents not erased, some blocks may not be
usable
... Erase from 0x607c0000-0x60800000: .
... Program from 0x03fbf000-0x03fff000 at 0x607c0000: .
RedBoot>
```

Load Root File System[2]

The default configuration of GESBC-9312 is using part of SDRAM as RAM disk for Linux root file system. The ramdisk image must be stored in the on-board FLASH memory and loaded by Redboot for the Linux kernel. The image must be loaded into dynamic memory before it can be stored in the on board FLASH memory. To load the ramdisk file to SDRAM, enter the following commands at the terminal console,

```
load –v –r –b 0x800000 –h tftp_host_ip ramdisk_file_name
```

where
-v : verbose
-r : binary format
-b : base address in memory

for serial port download, the command is,

```
load –v –r –b 0x800000 –m ymodem
```

Immediately after entered the above serial download command, start Y-Modem transfer on the terminal program, the download process should start.

The above commands will load ramdisk file into on board SDRAM. To store the image into non-volatile FLASH memory, use the following command,

```
fis create –b 0x800000 –l <ramdisk_file_length> ramdisk_file_name
```

where
-b : is the memory base address
-l : is the ramdisk size. It can be calculated by subtracting the end address from the base

[2] The host computer should have *tfpt* server running and Linux kernel and file system file stored in the *tftp* root directory.

address from the Redboot response when loading the ramdisk file.
The ramdisk_file_name can be any arbitrary name.

To verify the image has been stored correctly in the FLASH memory, use the following command,

```
fis list
```

Load Linux Kernel

The kernel image must be loaded into dynamic memory before it can be stored in the on-board FLASH memory. To load Linux kernel, issue the following command at the terminal console connected to the GESBC-9312 board,

```
load –v –r –b 0x80000 –h tftp_host_ip kernel_image_name
```

where
-v : verbose
-r : binary format
-b : base address in memory

The above command will load Linux kernel image file into on board SDRAM. To store the image into non-volatile FLASH memory, use the following command,

```
fis create –b 0x80000 –l <kernel_image_length> kernel_image_name
```

where
-b : is the memory base address
-l : is the kernel image size. It can be calculated by subtracting the end address from the base address from the Redboot response when loading the kernel image file.
The kernel_image_name can be any arbitrary name.

To verify the image has been stored correctly in the FLASH memory, use the following command,

```
fis list
```

Multiple kernel images or root file systems can be stored in the on-board FLASH memory when memory space permits.

Chapter 5 – Linux Development Tools

Overview

This chapter provides a brief introduction to development tools that are available for the EP9312 System-on-a-Chip processor. The central processing core on the EP9312 is a 200 MHz ARM920T processor. The ARM920T RISC processing core is supported through various toolsets available from third party suppliers. The typical toolset required for the code development is a compiler, assembler, linker and a source-level code debugger. Code debugging is supported via the on-chip JTAG interface.

Linux Development Tool Chain

The Linux development tool chain is included in the CD ROM come with the GESBC-9312 board. A host PC running Linux operating system is required to run the development tools. This guide assumes user had basic Linux or Unix application development knowledge.

Host Computer Requirement

The host PC should run Redhead, SuSe, or other Linux distribution, a RS-232 serial port, at least 500MB free disk space, and a terminal program such as minicom.

Hardware Connection

A null modem cable is required to connect GESBC-9312 to the host computer.

Install Linux Development Tool Chain

The ARM Linux Development Tool chain can be installed in any directory on the host system. The following example uses /usr/local/arm as the installing directory for the ARM Linux Development Tool Chain.

1. Create the tool chain directory and untar the tool chain

    ```
    cd /usr/local
    mkdir arm
    cd arm
    tar jxvf /cdrom/cross-3.3.tar.bz
    ```

    Your mounting point of CD-ROM maybe different

2. Set up the directory path variable

   export PATH=/usr/local/arm/3.3/bin:$PATH

   The above command can be included in the shell resource file so it is executed every time you login. For bash shell, a good place to put is in .bashrc in your home directory.

Compile Linux Kernel

The GESBC-9312 is shipped with Linux kernel version 2.4.21 with ARM patch and Cirrus Logic specific patch. User can download other version of Linux kernels from ftp://ftp.arm.linux.org.uk/pub/armlinux.

Prepare Linux Kernel source

Untar the Linux kernel, unzip the patches. And applying patches by executing the following commands,

```
tar jxvf linux-2.4.21.tar.bz2
bunzip2 linux-2.4.21.armk1.bz2
bunzip2 linux-2.4.21.armk1-cirrus.bz2
ln –s linux-2.4.21 linux
cd linux
patch –p1 < ../linux-2.4.21.armk1
patch –p1 <../linux-2.4.21.armk1-cirrus
```

Configure Linux Kernel

In the Linux kernel directory, executing the following commands,

```
make distclean
make ep9312_config
```

If no error message, proceed by

```
make oldconfig
make menuconfig
```

If problem occurs, make sure the default PATH variable is set to the correct tool chain directory

Compile Kernel

Once Linux kernel has been configured, it can be compiled using following commands

```
make dep
make    (or make zImage)
```

The Linux kernel should compile without error and the image file will be created.

Chapter 6 – Using Windows CE

Overview

This chapter provides information regarding how to use Windows CE with the GESBC-9315 Board. The information is only intended to provide a starting point to use the GESBC-9315. For more detailed information regarding Windows CE, please see Microsoft documents.

Hardware Requirement

The following tools and hardware are needed to build and download the Cirrus Logic EP931x Windows CE .net BSP the GESBC-9315 Board:

- Download.exe for the EP93xx
- GESBC-9315
- Ethernet Network with a DHCP server
- Null Modem, crossover Cable
- USB Keyboard or PS/2 Keyboard
- USB Mouse or Serial Mouse
- Windows 2000 SP3 or Windows XP
- Windows CE 4.2 Platform Builder with ARM4I support installed or Windows CE5.0 Platform Builder with EP931X BSP from Third Party Solutions disk

The PC should have the following specification

- 1.6Ghz Processor
- 10 Gig of Free Hard Drive Space before Platform builder is installed
- 256 Meg of RAM (512 MB recommended)
- Windows 2000 SP2 or Windows XP
- Network connection
- Serial port

Steps to Install EP931x BSP

The following are the steps to install the BSP for Windows CE 4.2 for EP931x BSP.

1. Unzip platform\ep931x from bsp_x-x-x.zip to %WINCEROOT%\PLATFORM\
2. Run Platform Builder.
3. Go to File->Manage Catalog Features.

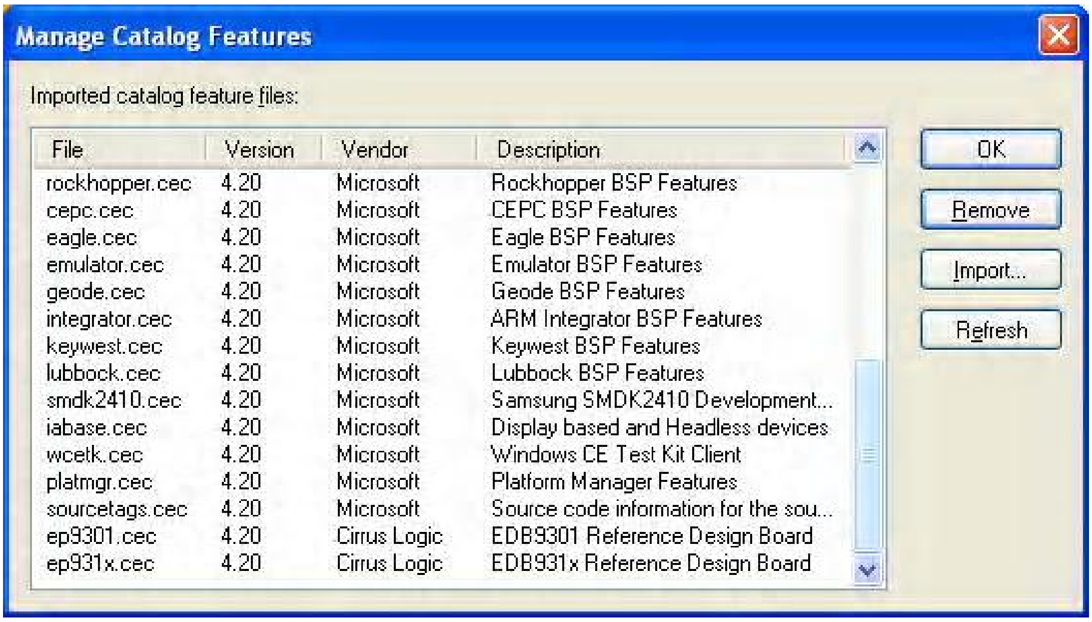


Figure 3 Manage Platform Builder Components

4. Click on the Import button.
5. Import the BSP from %WINCEROOT%\platform\ep931x\bspfiles\ep931x.cec.
6. Click on Refresh.
7. Close the Manage Catalog Features Window.

Setting up Windows CE Build Environment

1. Select File->New Platform.
2. Click Next.

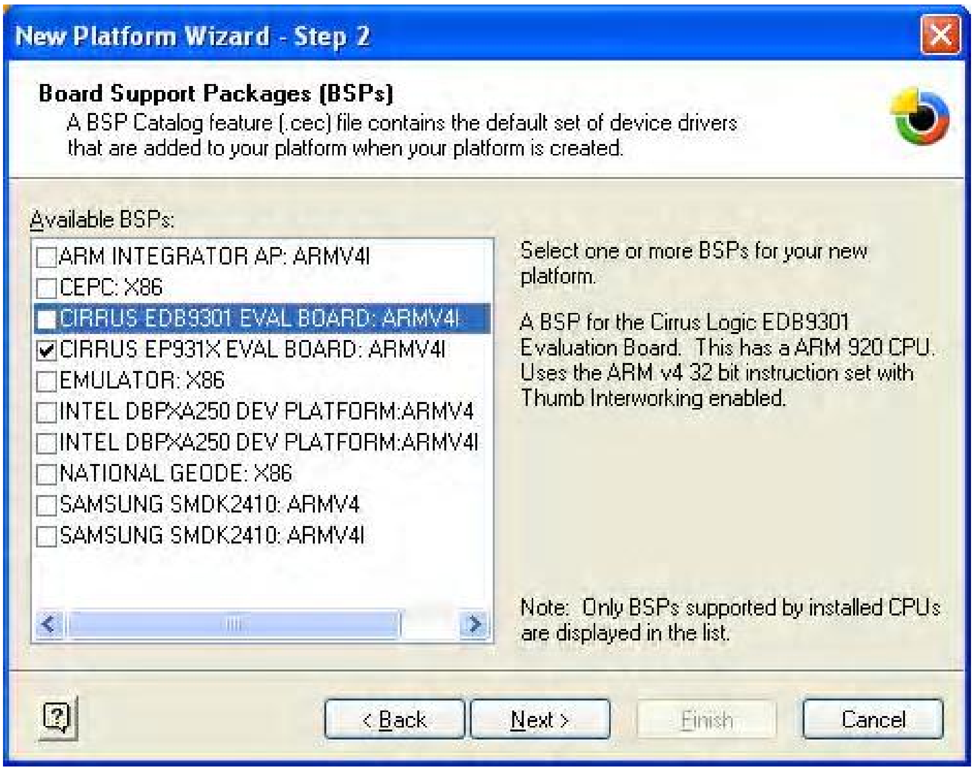


Figure 4 – Platform Setup (Page 1)

3. Select the EP931x Board Support Package (BSP) as shown in Figure 4.
4. Select Industrial Controller or whatever closely matches your platform as shown in Figure 5.

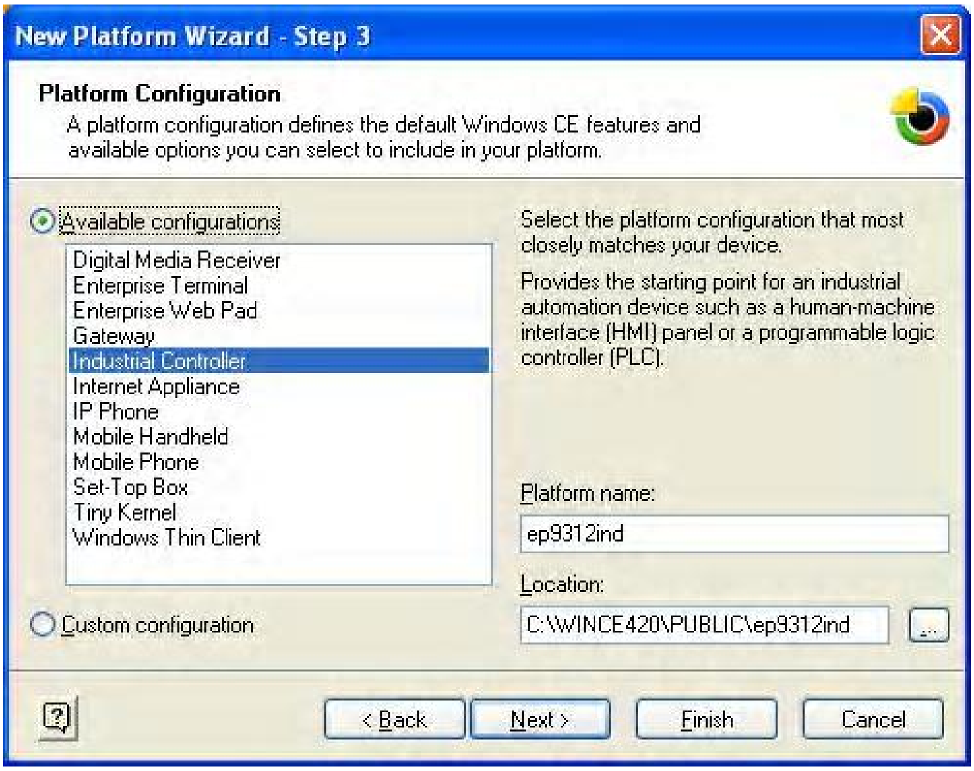


Figure 5 – Platform Setup (Page 2)

5. Then click on the "Finish" Button.
6. In the Catalog Window add and remove device drivers so that it looks like Figure 4. To remove a Device Driver/Feature select the item, and hit the delete key. To add a Device Driver/Feature right click the item in the Catalog Windows, and select "Add to Platform" from the Pop-Up Menu.

Remove the following items from the build:

- Local Area Networking (Lan) Devices. We are using the debug Ethernet driver.
- PCMCIA (PCCard) (PCMCIA should be kept for ep9315)

Note that PCI is shown in Figure 4 because it is required to build. PCI is not supported in the EP9312/EP9315 boards. All of the PCI routines are stubbed out.

Note: You cannot use debug Ethernet with the Ethernet NDIS driver. If you do your system will crash.

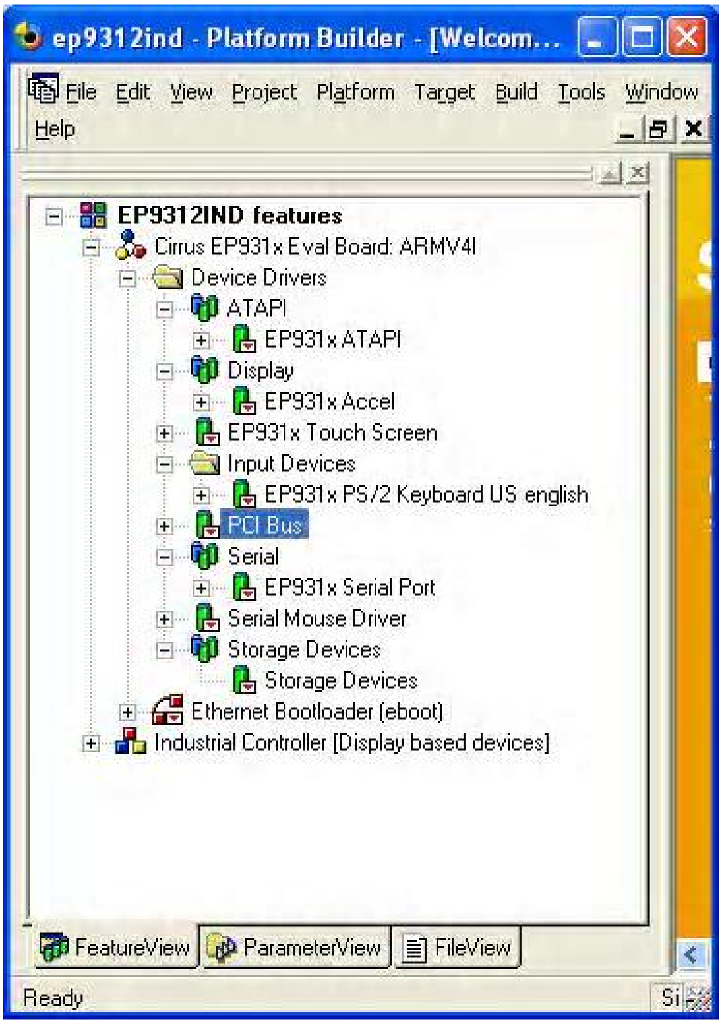


Figure 6 – Platform Features

7. Open %WINCEROOT%\platform\ep931x\inc\options.h in a file editor. A unique MAC address is required for every Ethernet device. Change the define "EBOOT_CS8950_MAC_ADDRESS" to some random unique number in the form of "{0xAABB, 0xCCDD, 0xEEFF}".
8. If you want to build Image for flash, you should select "Enable Image for Flash" in Platform->Settings->Build Options.
9. Select your default language in Platform->Settings->Locale.
10. Select release version as the active configuration.
11. Build the Image. Select Build->Build Platform. This will take a while.
12. After the build is performed, check to make sure that no errors have occurred in the Build Window.

Flashing Eboot

Eboot is a program that is in Flash memory. It allows a developer to download a Windows CE Image quickly into memory using Ethernet. Windows CE images can be large for example the image that we are building is about 10 Meg.

Eboot has been built in the last section. To program it into flash the following steps are needed.

1. Select Build->Open Build Release Directory.
2. Hook up serial port to UART 1 and the other end to PC.
3. Set jumper JP9 to 1 and 2 (factory default is 2 and 3).
4. Run "download.exe eboot.nb0"
5. Power up board. Download program should show that flash is being erased and that the program is being downloaded.
6. Wait until the download is finished.
7. Power down the board.
8. Put the jumper back to it's default position.

Testing Eboot and Download the Windows CE Image.

1. Make sure Jumper JP9 pin 2 and 3 are closed.
2. Hook up a null modem cable from UART 3 on the EP9312 board to the computer.
3. Hook up the Ethernet to a DHCP serviced Ethernet Network.
4. Open up a terminal program like HyperTerminal.
5. Set the serial port settings to:
   - 38400 Bits per Second
   - 8 Data Bits
   - No Parity
   - 1 Stop Bits
   - No Flow Control
6. Turn on the Board. You should get something similar to the output below:

```
Microsoft Windows CE Ethernet Bootloader Common Library Version 1.0 Built Apr 14
 2003 18:44:43
Copyright (c) 2000-2001  Microsoft Corporation

============================================================
Ep931x Windows CE Ethernet Bootloader
(Built on Apr 14 2003 18:44:49)
============================================================
Downloading BIN file using Ethernet.
CS8950DMAInit:  dwPhysicalAddr = 0x10000, dwVirtualAddr= 0x10000 dwSize = 0x3000
0
Card Type = CS8950, Address = 0x80010000.
CS8950Init:  pbBaseAddress = 0x80010000
CS8950Init:  pChip = 0x605C0, pChip->pData = 0x5FB08
System ready!
Preparing for download...
INFO: Using device name: 'NEXUS4660'
InitDHCP():: Calling ProcessDHCP()
ProcessDHCP()::DHCP_INIT
Got Response from DHCP server, IP address: 198.61.92.55
```

```
ProcessDHCP()::DHCP IP Address Resolved as 198.61.92.55, netmask: 255.255.255.0
Lease time: 259200 seconds
Got Response from DHCP server, IP address: 198.61.92.55
No ARP response in 2 seconds, assuming ownership of 198.61.92.55
+EbootSendBootmeAndWaitForTftp
Sent BOOTME to 255.255.255.255
```

Figure 7 - Eboot Starting Messages

7. Go to Target->Configure Remote Services in Platform Builder as shown in Figure 8. Set the "Download" and "Kernel Transport" List Boxes to Ethernet.

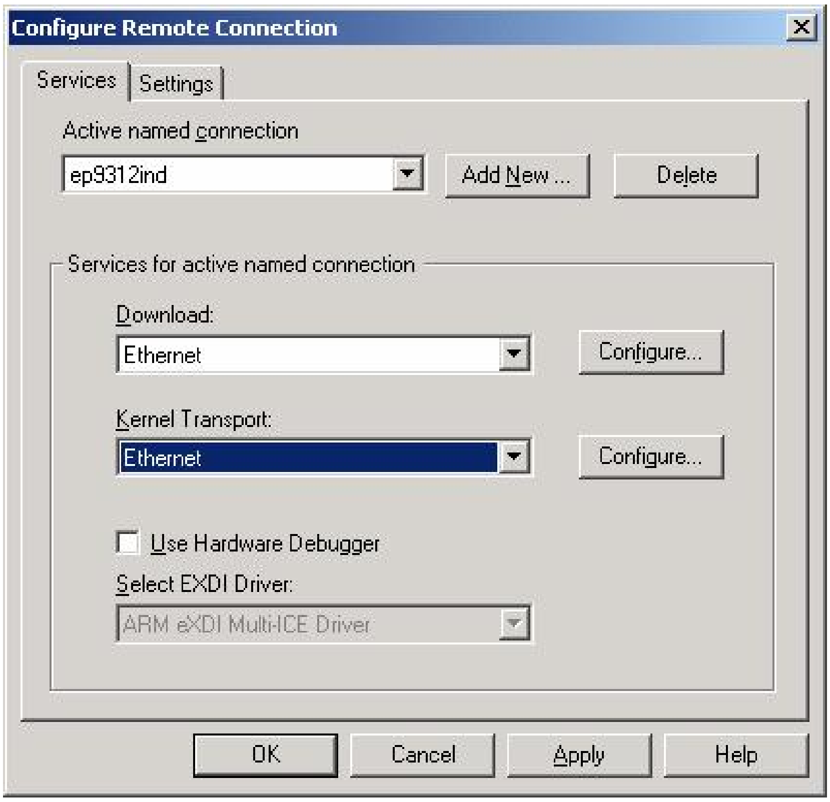


Figure 8 - Configure Remote Services

8. Push the configure button. You should see the your boards name in the "New Devices" List. Highlight the boards name and click on the button with the arrows in Figure 9.

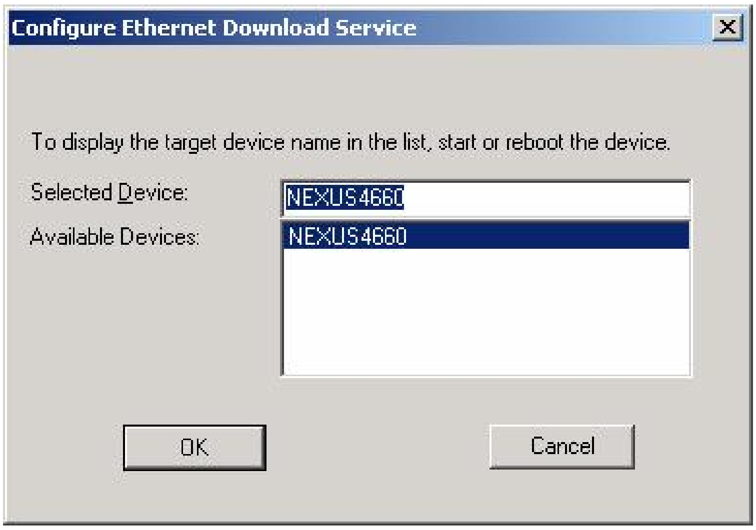


Figure 9 - Configure Ethernet Remote Services

Booting Windows CE

1. If you have not done so, connect USB keyboard on the board. Bring Up Board. Plug the Serial Mouse in to UART 1 or USB mouse in to a free USB connection.
2. After performing the steps in previous section, select Target->Download Image on the Platform Builder Menu Bar.

You should see a dialog box with a downloading pop up as shown in Figure 10. Also check the serial port console window to make sure that the download is occurring.

Figure 10 - Downloading Windows CE

After waiting about 10-20 seconds, the download will be completed. You should see messages out the Debug Window of Platform Builder that are similar to Figure 11.

```
Kernel debugger is waiting to connect with target.
      0 PID:0 TID:0 Processor = 199Mhz, Bus = 99Mhz
      0 PID:0 TID:0 Booting Windows CE version 4.20 for (ARM)
      0 PID:0 TID:0
Old or invalid version stamp in kernel structures - starting clean!
      0 PID:0 TID:0 Configuring: Primary pages: 8137, Secondary pages: 0,
Filesystem pages = 4068
      0 PID:0 TID:0
Booting kernel with clean memory configuration:
      0 PID:0 TID:0 Memory Sections:
      0 PID:0 TID:0 [0] : start: 82035000, extension: 00002000, length:
01fc9000
      0 PID:0 TID:0 Sp=ffffc7cc
    184 PID:3fff002 TID:3fe773e OEMInterruptEnable: Interrupt #24
    444 PID:c3fdf23e TID:c3fe4fca FileSystem Starting - starting with clean
file system
    844 PID:c3fdf23e TID:c3fe4fca Initobj : Error, continuing (1)...
    852 PID:c3fdf23e TID:c3fe4fca Initobj : Error, continuing (1)...
    915 PID:3fff002 TID:c3fe463e InitializeJit

Welcome to the Windows CE Shell. Type ? for help.

   2108 PID:83fccd8e TID:83fbfc72 Shell: No extension DLLs found
   3610 PID:3fbf522 TID:3fe4f46
                                        *
                                        *
```

Figure 11 - CETerm Output when CE Starts

You should also get messages in the serial terminal window saying Kitl is connected properly.

```
Got EDBG_CMD_JUMPIMG
Got EDBG_CMD_CONFIG, flags:0x00000000
EBOOT: pCfgData = 0x5F14A, pCfgData->Flags
INFO: Jumping to image at 0x00201006...
Windows CE Kernel for ARM (Thumb Enabled) Built on Mar  2 2003 at
19:43:16
ProcessorType=0920  Revision=0
sp_abt=ffff5000 sp_irq=ffff2800 sp_undef=ffffc800 OEMAddressTable =
802010b0
InitClock...
+OEMKitlInit
+InitEther
CS8950DMAInit:  dwPhysicalAddr = 0x10000, dwVirtualAddr= 0xA0010000
dwSize = 0x3
0000
INFO: EDBG using Internal Ep931x controller.
CS8950Init:  pbBaseAddress = 0xB0010000
CS8950Init:  pChip = 0x8202B8E0, pChip->pData = 0x8201F868
Debug Ethernet card initialized, MAC Address:00:24:20:10:12:34
VBridgeInit()...TX = [16384] bytes -- Rx = [16384] bytes
```

```
Tx buffer [0xA202F960] to [0xA2033960].
Rx buffer [0xA202B940] to [0xA202F940].
VBridge:: NK add MAC: [0-24-20-10-12-34]
Using device name: NEXUS4660
Device NEXUS4660, IP 198.61.92.55, Port 981
Calling EdbgInitDHCP
-InitEther
-OEMKitlInit
Host connected
```

Figure 12- Serial Port Output when CE Starts

Wait a little bit more until the CEterm messages stop. At this point the monitor should show the typical Windows CE screen as in Figure 13 Windows CE screen.

Figure 13 Windows CE screen

Chapter 7 – Troubleshooting

This chapter provides Troubleshooting information. Search the entries in the Problem column in order to find the item that best describes your situation. Then perform the corrective action in the same row. If the problem persists, contact Glomation.