GLOMATION

GESBC-9302E

User's Manual

Embedded Single Board Computer

Table of Contents

List of Tables

Chapter 1 – Introducing the GESBC-9302E Single Board Computer

GESBC-9302E Overview

The GESBC-9302E is a low cost compact sized single board computer based on Cirrus Logic EP9302 processor. With a large peripheral set targeted to a variety of applications, the GESBC-9302E is well suited for industrial controls, digital media servers, audio jukeboxes, thin clients, set-top boxes, point-of-sale terminals, biometric security systems, and GPS devices will benefit from the EP9302's integrated architecture and advanced features.

Advanced Features

The heart of the GESBC-9302E is the EP9302 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 EP9302 is a high-performance, low-power RISC-based device built around a single ARM920T microprocessor core. The ARM920T on the EP9302 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. A two-port USB host and two UARTs are included as well. The list below summarizes the features of the GESBC-9302E.

- 200MHz Processor Core – ARM920T with MMU
- MaverickCrunch™ Math Engine
- MaverickKey™ Security
- 32 ~ 64 MB SDRAM
- 4~16MB NOR FLASH
- Optional 128MB ~ 2GB NAND FLASH
- Ethernet Media Access Controller (EMAC)
- 5 channel 12-bit Analog-to-Digital Converter (ADC
- Universal Asynchronous Receiver / Transmitters (UARTs) with RS-485 Support

- 2 USB Host Port
- Real-Time Clock with battery backup hookup
- Hardware Debug Interface
- Optional 8 channel 12 bit programmable gain A/D converter with 4 additional digital I/O port
- Optional 4 channel 12 bit D/A converter

Figure 1 below shows a picture of the GESBC-9302E Single Board Computer.

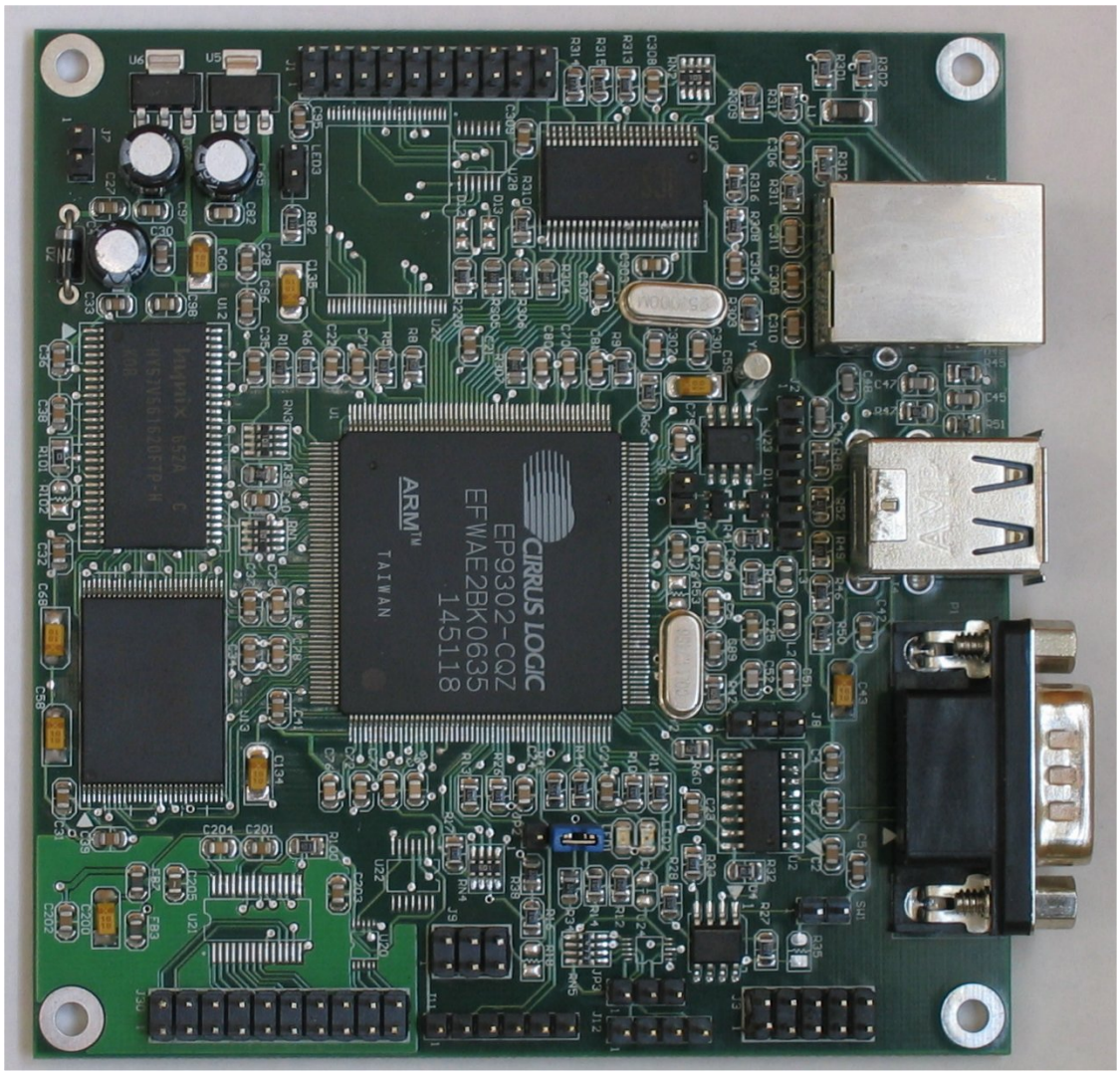


Figure 1. GESBC-9302E Single Board Computer

EP9302

The GESBC-9302E is shipped with the Cirrus Logic EP9302 processor. For more information regarding the EP9302 processor please see the EP9302 datasheet.

SDRAM

The GESBC-9302E is shipped with 32MBytes of SDRAM.

FLASH

The GESBC-9302E is shipped with 4MBytes of asynchronous Intel Strata-Flash. 128MB ~ 2GB NAND FLASH is available as an option.

USB

The GESBC-9302E is shipped with two USB host ports.

UART 1

The GESBC-9302E is shipped with a 9-pin interface.

UART 2

The GESBC-9302E is shipped with the 3 wire UART 2 interface.

Ethernet

The GESBC-9302E 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 EP9302 supports 1/10/100 Mbps transfer rates and interfaces to industry standard physical layer devices.

Chapter 2 – Getting Started

This chapter describes the GESBC-9302E 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-9302E Single Board Computer

- Operation
  - Describes how to operate the GESBC-9302E Single Board Computer

Assembly and Connections

In order to use the GESBC-9302E the user must first assemble and connect the peripherals to the GESBC-9302E, as described in the following procedure.

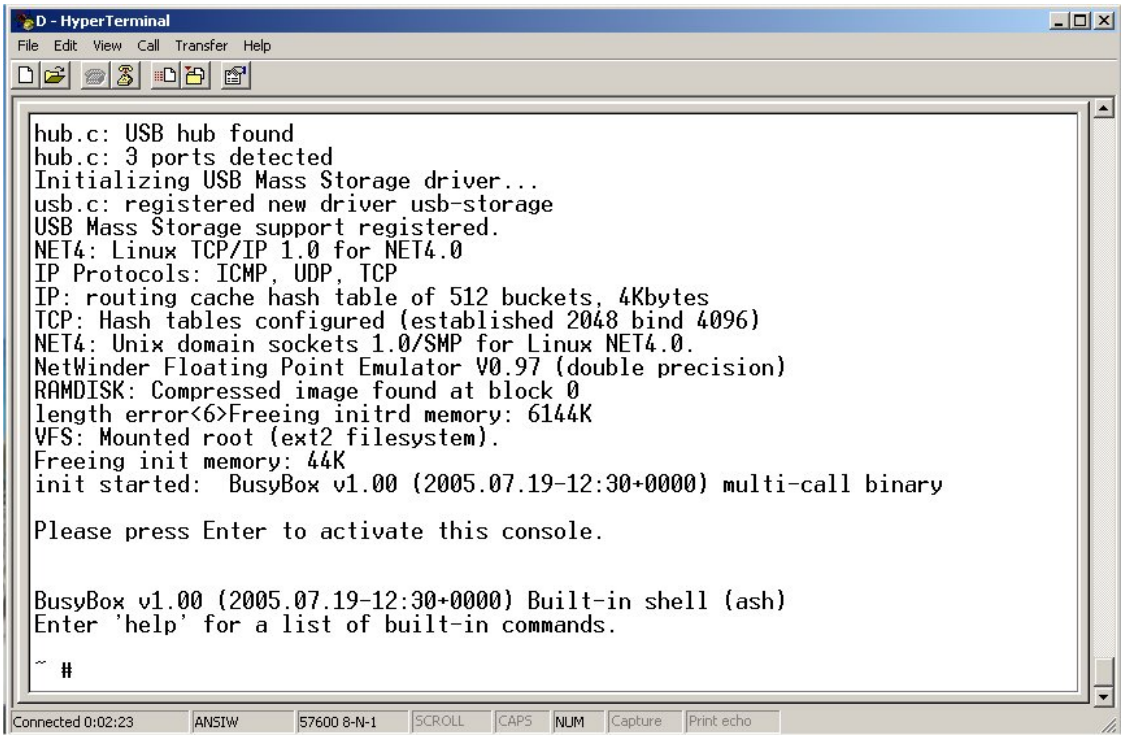
1. Place the GESBC-9302E on a static free surface.
2. Make sure all of the jumpers are in the factory default position. The unit is shipped in a factory default configuration. If the user is uncertain that the GESBC-9302E has the jumpers in the factory default configuration, please see the next section regarding board configuration.

3. Connect 5V regulated power supply to the board.
4. Connect null modem serial cable between GESBC-9302E 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-9302E. 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-9302E 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-9302E 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-9302E, debug information will be displayed on the terminal program. The following figure shows what this should look like.



Please see

Chapter 4 – Software Description for more details regarding the software functionality.

Configurations

Jumpers are used to configure the GESBC-9302E to operate in different mode. The following table lists all the settings for each jumper.

Table 1 System configuration

Jumper	Description
2	Boot mode: connect pin 1 and 2 - serial boot connect pin 2 and 3 - flash boot
SW1	Reset switch header
LED3	Power indicator LED header

Chapter 3 – GESBC-9302E Function Blocks

EP9302

The GESBC-9302E Single Board Computer uses the Cirrus Logic EP9302 as the core processor on this development board. The top-level features of EP9302 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
- 16 bit SDRAM Interface (Up To 4 Banks)
- 16 bit SRAM / FLASH / ROM Interface
- Serial EEPROM Interface
- 10 / 100 Mbps Ethernet MAC
- Two UARTs
- Two-port USB Host
- 5 channel 12 bit ADC
- SPI Port
- Serial Audio Interface
- JTAG Interface

More detailed information regarding the EP9302 processor can be found at www.cirrus.com and on the enclosed disk.

SDRAM

The EP9302 features a unified memory address model where all memory devices are accessed over a common address and data bus. The EP9302 can support a minimum of 1 to a maximum of 4 banks of 16-bit 66 or 100 MHz SDRAM. Additionally, the GESBC-9302E supports 32 SDRAM density. 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

The GESBC-9302E is shipped with 4 Mbytes of flash memory. The GESBC-9302E can be also ordered with optional 128MB ~ 2GB NAND FLASH.

USB

The GESBC-9302E Single Board Computer provides two USB host connections. The EP9302 USB host controller is configured for two root hub ports and features an integrated transceiver for each port. The EP9302 integrates two 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 standard double deck USB type A connector.

UART 1 and 2

The GESBC-9302E Single Board Computer is shipped with 2 3-wire UART interface. The UART 1 interface is provided via a standard DB-9 connector. The signal designation is listed in the following tables.

Table 2 P1 UART1 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	N/A

Table 3 J8 UART2 connector

Pin Number	Signal Name
1	RX
2	TX
3	GND

Ethernet

The GESBC-9302E Single Board Computer is shipped with support for a complete Ethernet interface. The EP9302 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 EP9302 supports 1/10/100 Mbps transfer rates and interfaces to industry standard physical layer devices. The GESBC-9302E is shipped with the ICS1893AF 100Base-X / 10Base-T Transceiver device which, along with a RJ45 connector, provides the physical layer interface.

SPI bus

The GESBC-9302E Single Board Computer is shipped with a SPI expansion bus header for peripheral expansion. The signal designation is listed in the following table.

Table 4 J11 SPI connector

Pin Number	Signal Name
1	SFRM
2	SSPRX
3	GND
4	SCLK
5	GND
6	SSPTX

On-chip A/D

The GESBC-9302E Single Board Computer is shipped with a 5 channel 12 bit on-chip A/D converter. The signal designation is listed in the following table.

Table 5 J3 On-chip A/D converter

Pin Number	Signal Name
1	GND
2	Channel 1
3	Channel 2
4	GND
5	GND
6	Channel 3
7	Channel 4
8	GND
9	GND
10	Channel 5

GPIO

The GESBC-9302E Single Board Computer provides 20 general purpose I/O signals for external use. The signal designation is listed in the following tables.

Table 6 J1 GPIO 1 connector

Pin Number	Signal Name	Pin Number	Signal Name
1	EGPIO8	2	EGPIO9
3	EGPIO10	4	EGPIO11
5	EGPIO12	6	EGPIO13
7	EGPIO14	8	EGPIO15
9	GPIO0 (CGPIO[0])	10	GPIO1 (HGPIO[5])

11	GPIO2 (HGPIO[4])	12	GPIO3 (HGPIO[3])
13	GPIO4 (HGPIO[2])	14	GPIO5 (FGPIO[1])
15	GPIO6 (FGPIO[2])	16	GPIO7 (FGPIO[3])
17	GND	18	GND
19	VDD3.3	20	VDD3.3

The GPIO3 and GPIO4 are used for optional A/D and D/A chip select when they are installed.

Table 7 J2 GPIO 2 connector

Pin Number	Signal Name	Pin Number	Signal Name
1	EGPIO2	2	EGPIO3
3	EGPIO4	4	EGPIO5
5	GND	6	VDD3.3

EGPIO 8 – 15 are connected directly to Port B of EP9302 CPU, EGPIO 2 – 5 are connected directly to Port A of EP9302 CPU. The GPIO 0 is mapped to Port C of EP9302 CPU. The GPIO 1 – 4 are mapped to Port H of EP9302 CPU which must be enabled in the DeviceCFG register at address 0x8093_0080 [1] . The GPIO 5 -7 are mapped to EP9302 port F.

RTC

In addition to the EP9302 on-chip RTC the GESBC-9302E contains an external real time clock with battery hook-up. The external RTC is DS1337/9 by Maxim-Dallas. The DS1337/9 interfaces the EP9302 CPU via the I2C bus. The DS1337 also is used to provide the 32KHz clock for the EP9302 CPU. When initializing/programming the DS1337 the 32KHz clock must be remain enabled to (default is enabled) allow the EP9302 CPU to proper functioning. The backup battery hook-up is listed in the following table,

Table 8 J5 RTC Battery Backup connector

Pin Number	Signal Name	Pin Number	Signal Name
1	3V	2	GND

[1] The EP9301/02 user guide does not provide clear documentation on the bit for port H. Please refer to page 145 of EP9315 user guide for Port H configuration bit.

JTAG

The GESBC-9302E Single Board Computer is shipped with a 6 pin connector that provides JTAG debug signals for the CPU. The JTAG provides the user with the ability to debug system level programs. The signal designation is listed in the following table.

Table 9 J9 JTAG connector

Pin Number	Signal Name	Pin Number	Signal Name
1	TRSTN	2	TDO
3	TDI	4	TMS
5	TCK	6	GND

Optional A/D and D/A

The GESBC-9302E 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 provides 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 10 J30 Optional 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	AIN1
9	AIN2	10	AIN3
11	AIN4	12	AIN5
13	AIN6	14	AIN7
15	AGND	16	AGND
17	AOUT0	18	AOUT1
19	AOUT2	20	AOUT3

Optional RS-485

The GESBC-9302E Single Board Computer provides support for optional full/half duplex RS-485 on UART2. The RS-485 signal is provided via J12. JP3 selects between full duplex and half duplex mode.

Table 11 J12 Optional RS-485

Pin Number	Signal Name	Pin Number	Signal Name

1	A1	2	B1
3	A2	4	B2

Table 12 JP3 RS-485 mode select

Pin Number	Signal Name
1	5V
2	Mode, high = half duplex mode, low = full duplex mode
3	GND

Reset Switch

The GESBC-9302E Single Board Computer provides a connector to connect to external reset switch.

Table 13 SW1 Reset connector

Pin Number	Signal Name
1	Reset (active low)
2	GND

Power Requirement

The GESBC-9302E Single Board Computer requires regulated 5v DC.

Table 14 J7 Power supply connector

Pin Number	Signal Name
1	5V DC
2	GND

Chapter 4 – Software Description

Overview

This chapter provides information regarding the software that is shipped with the GESBC-9302E 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-9302E board. The download utility provides a means to program a binary image into the flash memory on the GESBC-9302E.

GESBC-9302E Linux Code

The pre-programmed software provides the user with the opportunity to test some of the subsystems on the GESBC-9302E 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.

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.

USB Interface

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

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-9302E with a binary image. The following procedure will allow in-circuit programming of the flash memory via the EP9302 processor:

1) Power the board off.
2) Connect null-modem serial cable to UART1.
3) Set jumper 2 to connect pin 1 and 2 (JP2 factory default is pin 2 and 3).
4) Stop any program that might use the serial port that is connected to GESBC-9302E.
5) Run download utility (assuming download.exe located in same directory as binary image)
   download 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) Install jumper on pin 2 and 3 of JP2.
12) Power the board on.

Redboot

RedBoot provides a simple interface for loading operating systems and applications onto the GESBC-9302E 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-9302E board, the following message should be shown on the terminal console on the host PC connected to the GESBC-9302E board[2].

```
+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 EDB9301 Board (ARM920T) Rev A
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.
```

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

```
RAM: 0x00000000-0x04000000 available
FLASH: 0x60000000 - 0x60400000, 16 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 0x300000
>>
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>
```

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[3]

The default configuration of GESBC-9302E is using part of SDRAM as RAM disk for Linux root file system. The RAM disk 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 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-9302E board,

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

```
load –v –r –b 0x80000 –h host_ip_address 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 – Development Tools

Overview

This chapter provides a brief introduction to development tools that are available for the EP9302 System-on-a-Chip processor. The central processing core on the EP9302 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 ARM Linux development tool chain is widely available on the internet. The support page on the Glomation website http://www.glomationinc.com/support.html contains links to some recent version of tool chain. 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, Debian, 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-9302E 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. Download the Generic-arm_gcc-4.2.3-glibc-2.3.3.tar.bz2 cross tool from Glomation website and login as root,

   ```
   cd /
   tar jxvf /[file-path]/Generic-arm_gcc-4.2.3-glibc-2.3.3.tar.bz2
   ```

2. Set up the directory path variable

   ```
   export PATH=$PATH:/usr/local/arm/gcc-4.2.3-glibc-2.3.3/arm-
   ```

```
unknown-linux-gnu/
```

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-9302E is shipped with Linux kernel version 2.6.23.12. The patch for the main line Linux kernel source can be downloaded from http://www.glomationinc.com/support.html/.

Prepare Linux Kernel source

Download the main line kernel source from http://www.kernel.org. Untar the kernel source. Download the kernel patch for GESBC-9302E from http://www.glomationinc.com/support.html. Patch the kernel source using the following command,

```
patch –p1 < patch_file_path_and_name
```

Configure Linux Kernel

In the Linux kernel directory, executing the following commands,

```
make ARCH=arm CROSS_COMPILE=generic-arm-linux-gnu- 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    (or make zImage)
```

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

Native Application Development Over NFS

The network file system (NFS) can be used to do native kernel/application development. The include Debian based file system contains native compiler, debugger for ARM processor and many other utilities for fast application development.

The following steps outline the general procedure of mounting NSF on GESBC-9302E board.

1. Create a directory on server computer to house the file system for GESBC-9302E.

2. Download the Debian file system from Glomation website.
3. Unpack the included file system zip file in the newly created directory,
   `tar zxf <file path>/debian-etch-armbase.tar.gz`
4. Change the NFS server configuration to include the directory.
5. Power up GESBC-9302E and press CTRL-C at the terminal connected to the GESBC-9302E to stop the default redboot boot script. Load the on board kernel image from FLASH memory,
   `fis load zImage`
6. Start the Linux operating system by typing,
   `exec –c "root=/dev/nfs nfsroot=<server IP address>:/<NFS directory> ip=dhcp console=ttyAM0"`

The system is now ready to use for application development.

Chapter 6 – 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.