GLOMATION

Embedded Single Board Computer

GESBC-3130

User's Manual

Table of Contents

List of Tables

Chapter 1 – Introducing the GESBC-3130 Single Board Computer

GESBC-3130 Overview

The GESBC-3130 is a low cost compact sized single board computer based on NXP LPC-3130 processor. With a large peripheral set targeted to a variety of applications, the GESBC-3130 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.

Advanced Features

The heart of the GESBC-3130 is the LPC-3130 which is the one in a series of ARM926EJ-S-based processors. The NXP LPC3130 combines an 180 MHz ARM926EJ-S CPU core, high-speed USB 2.0 On-The-Go (OTG), up to 192 KB SRAM, NAND flash controller, flexible external bus interface, four channel 10-bit ADC, and a myriad of serial and parallel interfaces in a single chip targeted at consumer, industrial, medical, and communication markets.

The list below summarizes the features of the GESBC-3130.

- 180MHz Processor Core – ARM926EJ-S with MMU
- 32 MB SDRAM
- 128MB NAND FLASH
- 1 10/100 Mbps Ethernet port
- 4 channel 10-bit Analog-to-Digital Converter (ADC
- RS-232 Universal Asynchronous Receiver / Transmitters (UART)
- High speed USB OTG Port
- Real-Time Clock with battery backup
- Hardware Debug Interface
- SD/MMC Socket
- GPIO Ports
- 1 I2C Ports
- 1 SPI Port

Figure 1 below shows a picture of the GESBC-3130 Single Board Computer.

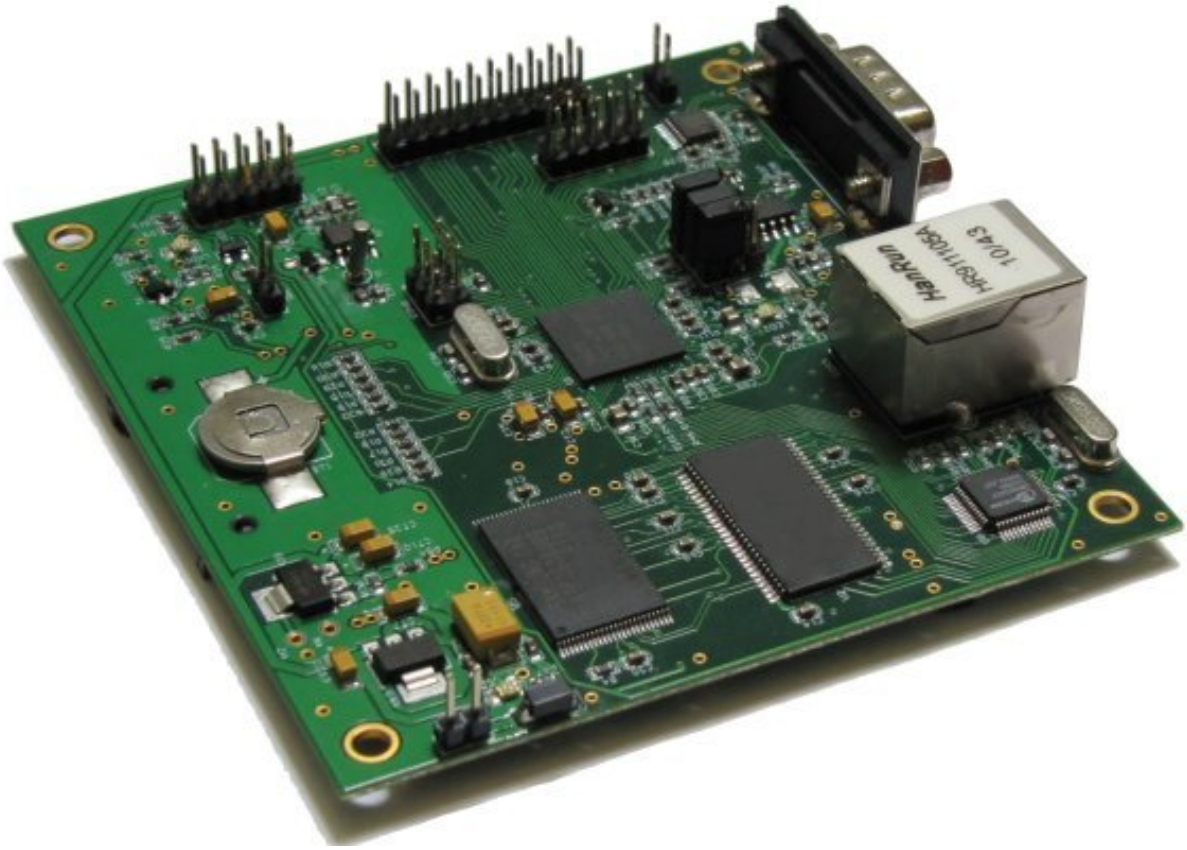


Figure 1. GESBC-3130 Single Board Computer

LPC-3130

The GESBC-3130 is shipped with the NXP LPC-3130 processor. For more information regarding the LPC-3130 processor please see the LPC-3130 datasheet.

SDRAM

The GESBC-3130 is shipped with 32MBytes of SDRAM.

FLASH

The GESBC-3130 is shipped with 128MB NAND FLASH.

USB

The GESBC-3130 is shipped with high speed USB OTG port.

UART

The GESBC-3130 is shipped with a full RS-232 interface with modem control signals

Ethernet

The GESBC-3130 is shipped with 10/100 BASE-T Ethernet port.

Chapter 2 – Getting Started

This chapter describes the GESBC-3130 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-3130 Single Board Computer

- Operation
  - Describes how to operate the GESBC-3130 Single Board Computer

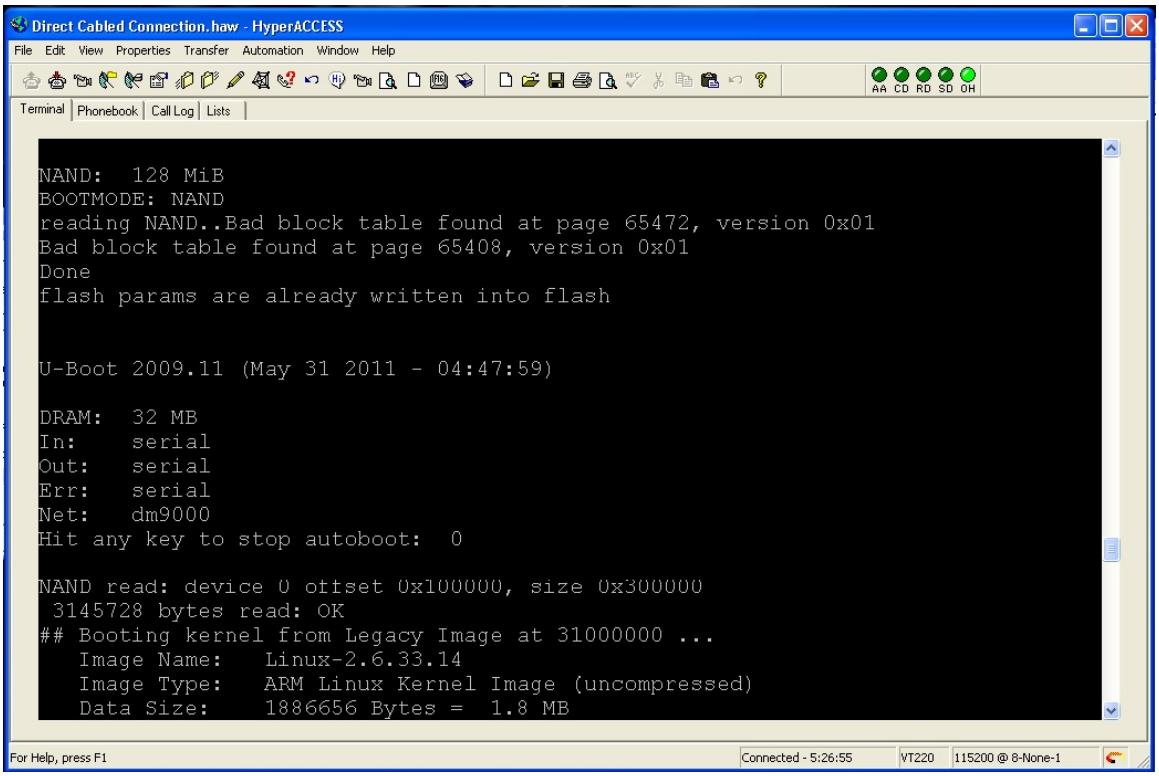
Assembly and Connections

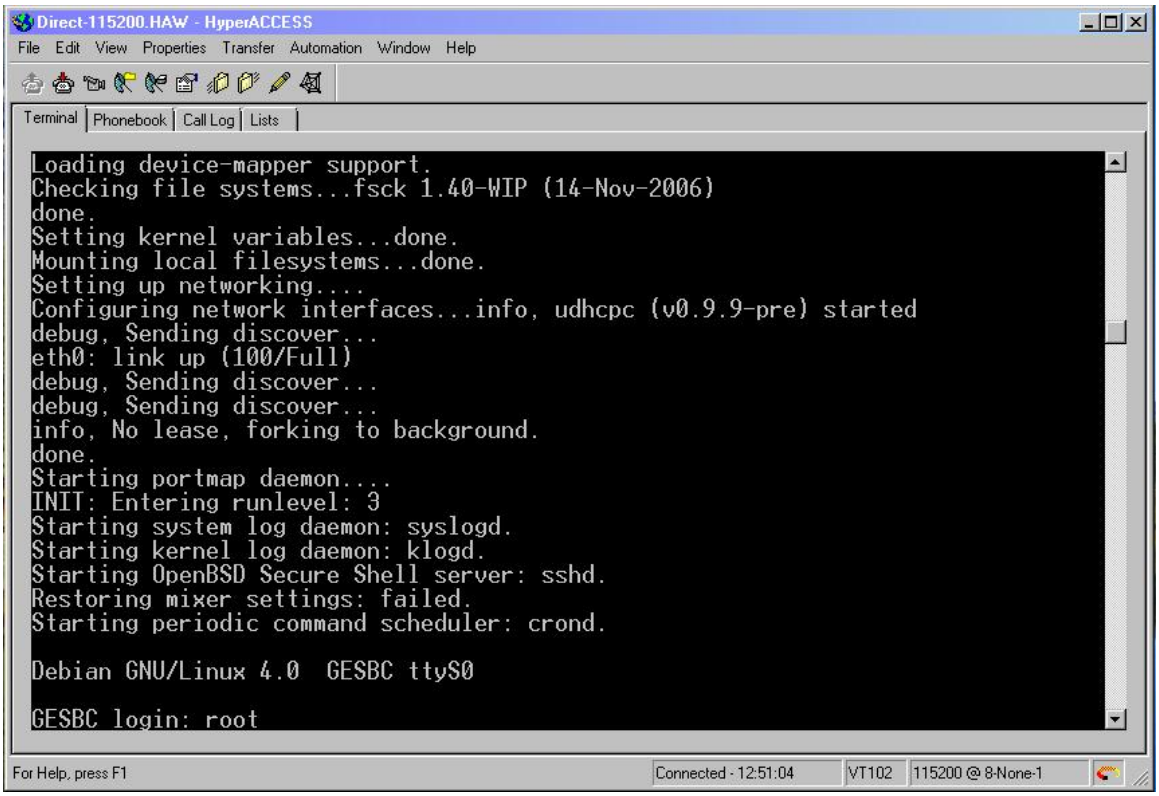
In order to use the GESBC-3130 the user must first assemble and connect the peripherals to the GESBC-3130, as described in the following procedure.

1. Place the GESBC-3130 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-3130 has the jumpers in the factory default configuration, please see the next section regarding board configuration.
3. Connect 5V regulated power supply to J1 on the board.
4. Connect null modem serial cable between GESBC-3130 debug port 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-3130. Configure the serial port with the following parameters: 115200 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

A few seconds after applying power to the GESBC-3130, debug information will be displayed on the terminal program. The following figures show what this should look like.


Direct Cabled Connection.haw - HyperACCESS
File Edit View Properties Transfer Automation Window Help
Terminal Phonebook Call Log Lists
NAND: 128 MiB
BOOTMODE: NAND
reading NAND..Bad block table found at page 65472, version 0x01
Bad block table found at page 65408, version 0x01
Done
flash params are already written into flash
U-Boot 2009.11 (May 31 2011 - 04:47:59)
DRAM: 32 MB
In: serial
Out: serial
Err: serial
Net: dm9000
Hit any key to stop autoboot: 0
NAND read: device 0 offset 0x100000, size 0x300000
3145728 bytes read: OK
Booting kernel from Legacy Image at 31000000 ...
Image Name: Linux-2.6.33.14
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 1886656 Bytes = 1.8 MB
For Help, press F1
Connected - 5:26:55
VT220
115200 @ 8-None-1


Direct-115200.HAW - HyperACCESS
File Edit View Properties Transfer Automation Window Help
Terminal Phonebook Call Log Lists
Loading device-mapper support.
Checking file systems...fsck 1.40-WIP (14-Nov-2006)
done.
Setting kernel variables...done.
Mounting local filesystems...done.
Setting up networking....
Configuring network interfaces...info, udhcpc (v0.9.9-pre) started
debug, Sending discover...
eth0: link up (100/Full)
debug, Sending discover...
debug, Sending discover...
info, No lease, forking to background.
done.
Starting portmap daemon....
INIT: Entering runlevel: 3
Starting system log daemon: syslogd.
Starting kernel log daemon: klogd.
Starting OpenBSD Secure Shell server: sshd.
Restoring mixer settings: failed.
Starting periodic command scheduler: crond.
Debian GNU/Linux 4.0 GESBC ttyS0
GESBC login: root
For Help, press F1
Connected - 12:51:04
VT102
115200 @ 8-None-1

Please see

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

Configurations

The GESBC-3130 has multiple boot mode. It can boot from SD/MMC card, USB port, UART, or on-board NAND FLASH. Jumpers are used to configure the GESBC-3130 to boot in different modes. The following table lists all the settings for each boot mode.

Table 1 Boot Mode Configuration

JP2	JP3	JP4	Boot Mode
L	L	L	NAND FLASH boot
L	H	H	SD/MMC card boot
H	H	L	UART boot
L	H	L	USB boot

Chapter 3 – GESBC-3130 Function Blocks

LPC-3130

The GESBC-3130 Single Board Computer uses the NXP LPC-3130 as the core processor on this development board. The top-level features of LPC-3130 processor are the following:

- ARM926EJ-S RISC Core Processor
- 180MHz
- 16Kbyte Instruction Cache
- 16 Kbyte Data Cache
- MMU
- 100 MHz System Bus
- 16 bit SDRAM Interface
- 8/16 bit SRAM / FLASH / ROM Interface
- Fast UART
- High Speed USB OTG Port with on-chip Physical
- 4 channel 10 bit ADC
- SPI Port
- 2 I2C Port
- Serial Audio Interface
- General Purpose I/O Pins (GPIO)
- JTAG Interface

More detailed information regarding the LPC-3130 processor can be found at www.NXP.com.

SDRAM

The GESBC-3130 is shipped with 32MB SDRAM. It occupies memory location from 0x3000-0000 to 0x31FF-FFFF.

FLASH

The GESBC-3130 is shipped with 128 Mbytes of NAND FLASH memory.

USB

The GESBC-3130 Single Board Computer provides one high speed USB OTG port. The USB On-The-Go block enables usage in both device mode and in host mode. This means that you can connect to a PC to exchange data, but also to another USB device such as a digital camera or MP3 player. The USB port are brought out by a USB type A/B mini connector.

RS-232 Port

The GESBC-3130 Single Board Computer is shipped with one fast RS-232 UART interface. The fast UART on the GESBC-3130 serves as a debug port by U-boot and Linux kernel. The UART connector is the DB-9 connector on GESBC-3130. The signal designation is listed in the following table.

Table 2 UART Port CON8 Connector on GESBC-3130

Pin Number	Signal Name	Pin Number	Signal Name
1	NC	2	RX
3	TX	4	NC
5	GND	6	NC
7	RTS	8	CTS
9	NC	10	N/A

I2C Bus, SPI Bus, PWM output

The GESBC-3130 Single Board Computer provides one I2C bus interface, one SPI bus, and one PWM output on connector CON2. The I2C bus on the CON2 is the I2C1 on the LPC-3130 processor.

Table 3 CON2 I2C bus, SPI bus, PWM output

Pin Number	Signal Name	Pin Number	Signal Name
1	SDA	2	SPI CS
3	SCL	4	SPI CLK
5	PWM output	6	SPI MISO
7	3.3V	8	SPI MOSI
9	GND	10	SPI CS_IN

Ethernet

The GESBC-3130 Single Board Computer is shipped with support for a complete Ethernet interface. The Media Access Controller (MAC) supports 1/10/100 Mbps transfer rates and interfaces to industry standard physical layer devices.

USB Port

The GESBC-3130 Single Board Computer is shipped with a high speed USB OTG port on a USB mini A/B USB OTG connector.

A/D Converter

The GESBC-3130 Single Board Computer provides 4 channel 10 bit A/D converter. A 6pin 2.54mm spacing header provides the 4 A/D input channel.

Table 4 CON9 A/D input

Pin Number	Signal Name
1	A/D 1
2	A/D 2
3	A/D 3
4	A/D 4
5	GND
6	GND

GPIO

The GESBC-3130 is shipped with 16 configurable general purpose input/output (GPIO) ports. Each GPIO port can be configured to have pull-up, pull-down or repeater mode. The signal assignment is listed in the following table

Table 5 CON1 GPIO

Pin	Signal	Pin	Signal
1	3.3V	2	3.3V
3	GPIO 3	4	GPIO 4
5	GPIO 5	6	GPIO 6
7	GPIO 7	8	GPIO 8
9	GPIO 9	10	GPIO 10
11	GPIO 11	12	GPIO 12
13	GPIO 13	14	GPIO 14
15	GPIO 15	16	GPIO 16
17	GPIO 17	18	GPIO 18
19	GND	20	GND

Real Time Clock (RTC)

The GESBC-3130 is shipped with a real time clock (RTC) with battery hook-up to provide accurate time keeping. The on-board battery holder accepts CR1225/CR1220 coin cell batteries.

JTAG

The GESBC-3130 Single Board Computer is shipped with a 10 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 6 CON7 JTAG Connector

Pin Number	Signal Name	Pin Number	Signal Name
1	3.3V	2	NC
3	NTRST	4	N_RESET_IN
5	TDI	6	TDO
7	TMS	8	NC
9	TCK	10	GND

Power Requirement

The GESBC-3130 Single Board Computer requires regulated 5V DC. The power supply should have minimum 250mA capacity.

Table 7 CON11 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-3130 Board. The software included with the board is U-boot boot loader, Linux kernel 2.6.33.14, and Debian distribution style compact root file system. The applications included provide access to all hardware functions on the GESBC-3130 board.

Data Storage on GESBC-3130

The default configuration of the GESBC-3130 Single Board Computer uses on board NAND FLASH for all data storage requirements, including boot strap code, boot loader, Linux kernel, and Linux file system.

The following table is the storage map on the NAND FLASH.

Table 8 NAND FLASH Storage Map

Start Address	Size	Usage
0x00000000	0x20000	Boot strap code
0x00020000	0x40000	U-boot
0x00060000	0x40000	U-boot primary environment storage range
0x000A0000	0x40000	U-boot secondary environment storage range
0x00100000	0x300000	Linux kernel
0x00400000	--	Root file system

GESBC-3130 Linux Code

The GESBC-3130 is shipped with Linux 2.6.33.14 kernel pre-installed. This software is programmed into the system FLASH located on the board prior to shipment. The Linux kernel is configured with all the device drivers included for the GESBC-3130 board.

U-boot

U-boot provides a simple interface for loading operating systems and applications onto the GESBC-3130 board. U-Boot uses a serial console for its input and output. The default serial port setting is 115200,8,N,1. It also supports the built-in Ethernet port and general flash programming.

The board is shipped with U-boot pre-installed. Please refer to U-boot user's manual regarding detailed information of U-boot.

U-boot Booting Linux

The following shows the default U-boot setup for booting Linux.

```
GESBC-3130 # printenv
bootcmd=run nand_boot
bootdelay=3
baudrate=115200
bootfile=uImage
loadaddr=0x31000000
rd_addr=0x32000000
usbtty=cdc_acm
ramargs=setenv bootargs console=ttyS0,115200n8 root=/dev/ram0 rw
ip=dhcp loglevel=7
nfsargs=setenv bootargs console=ttyS0,115200n8 root=/dev/nfs rw
nfsroot=${serverip}:${rootpath} ip=dhcp loglevel=7
rootpath=/tftpboot/arm
bootfile=uImage
usb_boot= usb start; fatload usb 0 $(loadaddr) $(bootfile); run
nandargs; bootm $(loadaddr)
uart_boot= loady; run nfsargs; bootm $(loadaddr)
uartram_boot= loady; loady $(rd_addr); run ramargs; bootm $(loadaddr)
$(rd_addr)
ethaddr=00:08:9e:d1:bd:6a
ethact=dm9000
nandargs=setenv bootargs console=ttyS0,115200n8 root=/dev/mtdblock2
rw rootfstype=jffs2 mtdparts=lpc313x_nand:1M(boot),3M(kernel),-(fs)
filesize=20423C
fileaddr=31000000
ipaddr=192.168.1.200
serverip=192.168.1.199
nand_boot=nand read $(loadaddr) 0x100000 0x300000;run nandargs; bootm
$(loadaddr)

Environment size: 916/131068 bytes
```

The `bootcmd` setting of the U-boot reads the Linux kernel from NAND FLASH at address 0x100000 to SDRAM at address 0x31000000 and start executing the kernel code at the same memory address. The NAND FLASH from 0x400000 and up is used for Linux root file system. The U-boot passes the MTD device partition setting to the Linux kernel via the `bootargs` environment variable.

Loading Linux Kernel and root File System

The U-boot boot-loader provides many ways to load Linux kernel and file system into FLASH memory. The loading by Ethernet network is shown here. User can consult U-boot manual for other methods of loading data.

After power on the GESBC-3130 board, stop the U-boot auto-execution by press any key. The following message should be shown on the terminal console on the host PC connected to the GESBC-3130 board.

The net work address and server address must be set before network transfer can take place. The following commands will set the SBC IP address and server IP address,

```
set ipaddr xxx.xxx.xxx.xxx
set serverip xxx.xxx.xxx.xxx
```

The server IP is the IP address where a TFTP server must be run. To load Linux kernel type in the following command,

```
t 0x31000000 uImage
```

The U-boot will load uImage file from the TFTP server whose IP address is specified by the `serverip` environment vairbale.

The NAND FLASH sectors must be erased first before new kernel image can be stored. The following command will erase the NAND FLASH sectors reserved for Linux kernel,

```
nand erase 0x100000 0x200000
```

The use the flowing command to store the kernel image from SDRAM to NAND FLASH,

```
nand write.jffs2 0x31000000 0x100000 0x200000
```

The following commands can be used to load root file system into the FLASH memory,

```
nand erase 0x400000 0x7c000000
t 0x30000000 rootfs.img
nand write.jffs2 0x30000000 x0400000 $(filesize)
```

Please be noted that the image is first loaded into the SDRAM and then stored into the FLASH memory. The image size can not exceed the available SDRAM on the board.

After the kernel and root file system have been updated the board can be simply reboot by recycle the power.

Chapter 5 – Development Tools

Overview

This chapter provides a brief introduction to development tools that are available for the LPC-3130 System-on-a-Chip processor. The central processing core on the LPC-3130 is a 180 MHz ARM926EJ-S processor. The ARM926EJ-S 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 available at Glomation website in the support page. 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-3130 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 cross compiler default directory /usr/local/arm as the installing directory for the ARM Linux cross compiler.

1. Login as root and untar the tool chain

```
cd /
tar jxvf /<cross compiler tar file directory>/ Generic-arm_gcc-4.2.3-
glibc-2.3.3.tar.bz2
```

2. Set up the directory path variable

```
export PATH=/usr/local/arm/gcc-4.2.3-glibc-2.3.3/arm-unknown-linux-
gnu/bin:$PATH
```

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-3130 is shipped with Linux kernel version 2.6.33.14. The patch for the kernel source tree is available at Glomation website in the support page.

Prepare Linux Kernel source

Obtain the kernel source 2.6.33.14 from http://www.kernel.org. Untar the Linux kernel,

```
tar xjf  linue-2.6.33.14.bz2
```

Obtain the kernel patch from http://ics.nxp.com/support/software/lpc313x.bsp.linux/. Patch the kernel source with the following command,

```
patch –p1 < /<patch-file-directory-path>/patch_file_name
```

Configure Linux Kernel

The GESBC-3130 can use the default configuration file for the EA3130 evaluation board.

```
make ARCH=arm CROSS_COMPILE=arm-unknown-linux-gnu- ea313x_defconfig
```

If additional configuration is required, executing the following command in the Linux kernel directory,

```
make ARCH=arm CROSS_COMPILE=arm-unknown-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 command,

```
make ARCH=arm CROSS_COMPILE=arm-unknown-linux-gnu- UImage
```

The U-boot utility mkimage is required to make the U-boot formatted kernel image files.

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.