



Embedded Single Board Computer

**GESBC-9G20i**

**User's Manual**

Preliminary

## Table of Contents

Chapter 1 – Introducing the GESBC-9G20i Single Board Computer .....	4
GESBC-9G20i Overview.....	4
Advanced Features.....	4
AT91SAM9G20.....	5
SDRAM .....	5
FLASH.....	5
USB.....	5
UART 1.....	5
UART 2.....	6
DEBUG Port.....	6
Ethernet.....	6
Chapter 2 – Getting Started.....	7
Assembly and Connections.....	7
Operation.....	7
Chapter 3 – GESBC-9G20i Function Blocks .....	10
AT91SAM9G20.....	10
SDRAM .....	10
FLASH.....	10
RS-232 Port 0, 1, and 2.....	10
RS-485 .....	11
I2C Bus .....	11
Ethernet.....	12
USB Port.....	12
I/O Block.....	12
4 x 20 Character LCD Port J30.....	13
RTC.....	13
Optional A/D.....	14
JTAG.....	14
Power Requirement.....	15
Chapter 4 – Software Description.....	16
Overview.....	16
Data Storage on GESBC-9G20i.....	16
GESBC-9G20i Linux Code .....	16
U-boot .....	16
U-boot Booting Linux.....	16
Loading Linux Kernel and root File System.....	17
Chapter 5 – Development Tools .....	19
Overview.....	19
Linux Development Tool Chain .....	19
Chapter 6 – Troubleshooting .....	21

## List of Tables

Table 1 UART Port P0 Connector on GESBC-9G20i.....	11
Table 2 UART Port P1 Connector .....	11
Table 3 J9 I2C bus .....	11
Table 4 J8 I/O Expansion.....	12
Table 5 J30 LCD.....	13
Table 6 J12 Optional A/D.....	14
Table 7 J20 JTAG Connector .....	14
Table 8 J1 Power Supply Connector.....	15
Table 9 NAND FLASH Storage Map.....	16

Preliminary

## Chapter 1 – Introducing the GESBC-9G20i Single Board Computer

### GESBC-9G20i Overview

The GESBC-9G20i is a low cost compact sized single board computer based on Atmel AT91SAM9G20 processor. With a large peripheral set targeted to a variety of applications, the GESBC-9G20i 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.

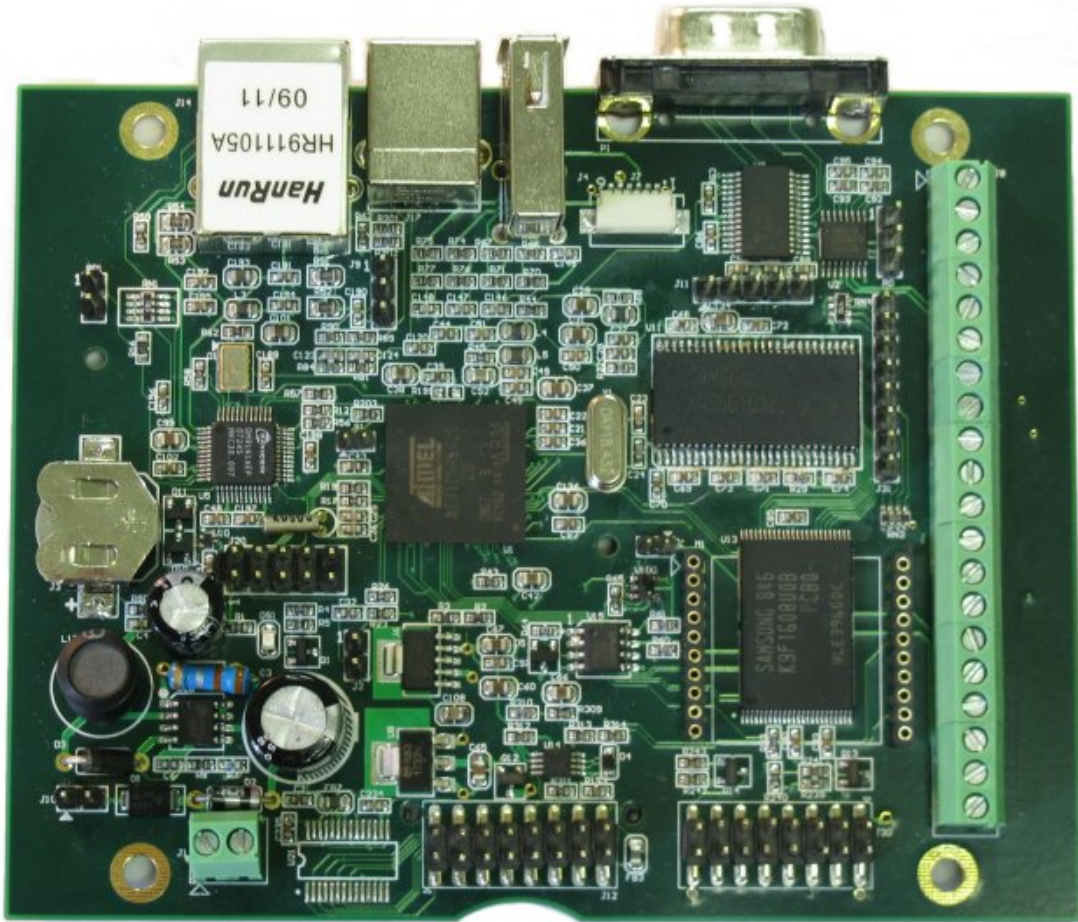
### Advanced Features

The heart of the GESBC-9G20i is the AT91SAM9G20 which is the one in a series of ARM926EJ-S-based processors. The AT91SAM9G20 microcontroller features DSP Instruction Extensions, ARM Jazelle® Technology for Java® Acceleration. It has separate 32 Kbyte instruction and data caches with write buffer. The ARM926EJ-S on the AT91SAM9G20 functions with a maximum operating clock rate of 400MHz and a power usage between 20mW and 80mW (dependent upon clock speed). The ARM core operates from a 1V supply while the I/O operates at 3.3V. The low power consumption makes it an idea platform for battery operated applications.

The list below summarizes the features of the GESBC-9G20i.

- 400MHz Processor Core – ARM926EJ-S with MMU
- 32~ 64 MB SDRAM
- 128MB ~ 1GB NAND FLASH
- 1 10/100 Mbps Ethernet port
- 4 channel 10-bit Analog-to-Digital Converter (ADC)
- 3 RS-232 Universal Asynchronous Receiver / Transmitters (UARTs)
- 1 RS-485 Port
- 2 USB Host Port
- 1 USB Device Port
- Real-Time Clock with battery backup
- Hardware Debug Interface
- SD/MMC Socket
- 1 I2C Port
- 1 SPI Port

Figure 1 below shows a picture of the GESBC-9G20i Single Board Computer.



**Figure 1. GESBC-9G20i Single Board Computer**

### **AT91SAM9G20**

The GESBC-9G20i is shipped with the Atmel AT91SAM9G20 processor. For more information regarding the AT91SAM9G20 processor please see the AT91SAM9G20 datasheet.

### **SDRAM**

The GESBC-9G20i is shipped with 32MBytes of SDRAM.

### **FLASH**

The GESBC-9G20i is shipped with 128MB NAND FLASH.

### **USB**

The GESBC-9G20i is shipped with two USB host ports.

### **UART 1**

The GESBC-9G20i is shipped with a RS-232 interface with RTS and CTS hardware flow control signals

## **UART 2**

The GESBC-9G20i is shipped with the 3 wire UART 2 interface.

## **DEBUG Port**

The GESBC-9G20i is shipped with the 3 wire serial debug port.

## **Ethernet**

The GESBC-9G20i 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 AT91SAM9G20 supports 1/10/100 Mbps transfer rates and interfaces to industry standard physical layer devices.

Preliminary

## Chapter 2 – Getting Started

This chapter describes the GESBC-9G20i 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-9G20i Single Board Computer
- Operation
  - Describes how to operate the GESBC-9G20i Single Board Computer

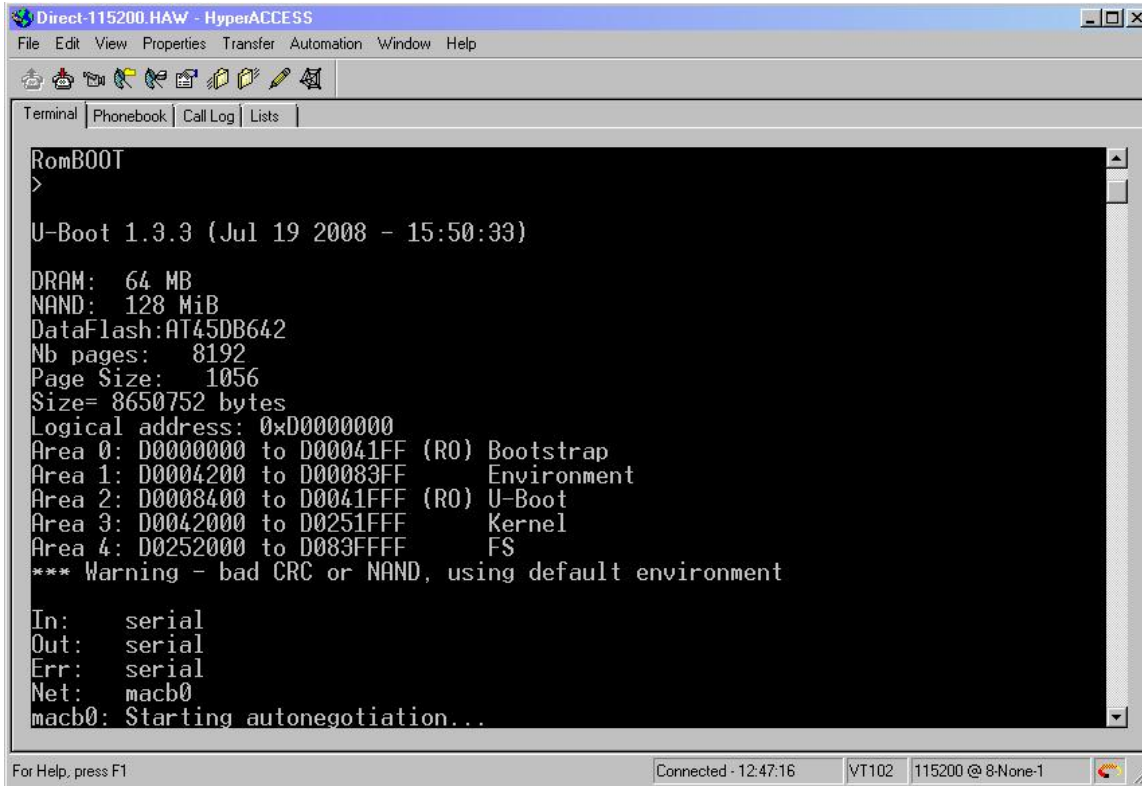
### Assembly and Connections

In order to use the GESBC-9G20i the user must first assemble and connect the peripherals to the GESBC-9G20i, as described in the following procedure.

1. Place the GESBC-9G20i 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-9G20i has the jumpers in the factory default configuration, please see the next section regarding board configuration.
3. Connect 7.5 ~ 18V DC power supply to the board through J1.
4. Connect null modem serial cable between GESBC-9G20i debug port P0 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-9G20i. 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-9G20i, debug information will be displayed on the terminal program. The following figures show what this should look like.



The screenshot shows a terminal window titled "Direct-115200.HAW - HyperACCESS". The terminal output displays the U-Boot boot process. It starts with "RomBOOT" and "U-Boot 1.3.3 (Jul 19 2008 - 15:50:33)". It lists hardware details: "DRAM: 64 MB", "NAND: 128 MiB", "DataFlash: AT45DB642", "Nb pages: 8192", "Page Size: 1056", and "Size= 8650752 bytes". It then shows memory areas: "Area 0: D0000000 to D00041FF (RO) Bootstrap", "Area 1: D0004200 to D00083FF Environment", "Area 2: D0008400 to D0041FFF (RO) U-Boot", "Area 3: D0042000 to D0251FFF Kernel", and "Area 4: D0252000 to D083FFFF FS". A warning message follows: "\*\*\* Warning - bad CRC or NAND, using default environment". At the bottom, it shows "In: serial", "Out: serial", "Err: serial", "Net: macb0", and "macb0: Starting autonegotiation...". The status bar at the bottom indicates "Connected - 12:47:16", "VT102", and "115200 @ 8-None-1".

```
Direct-115200.HAW - HyperACCESS
File Edit View Properties Transfer Automation Window Help

Terminal | Phonebook | Call Log | Lists |

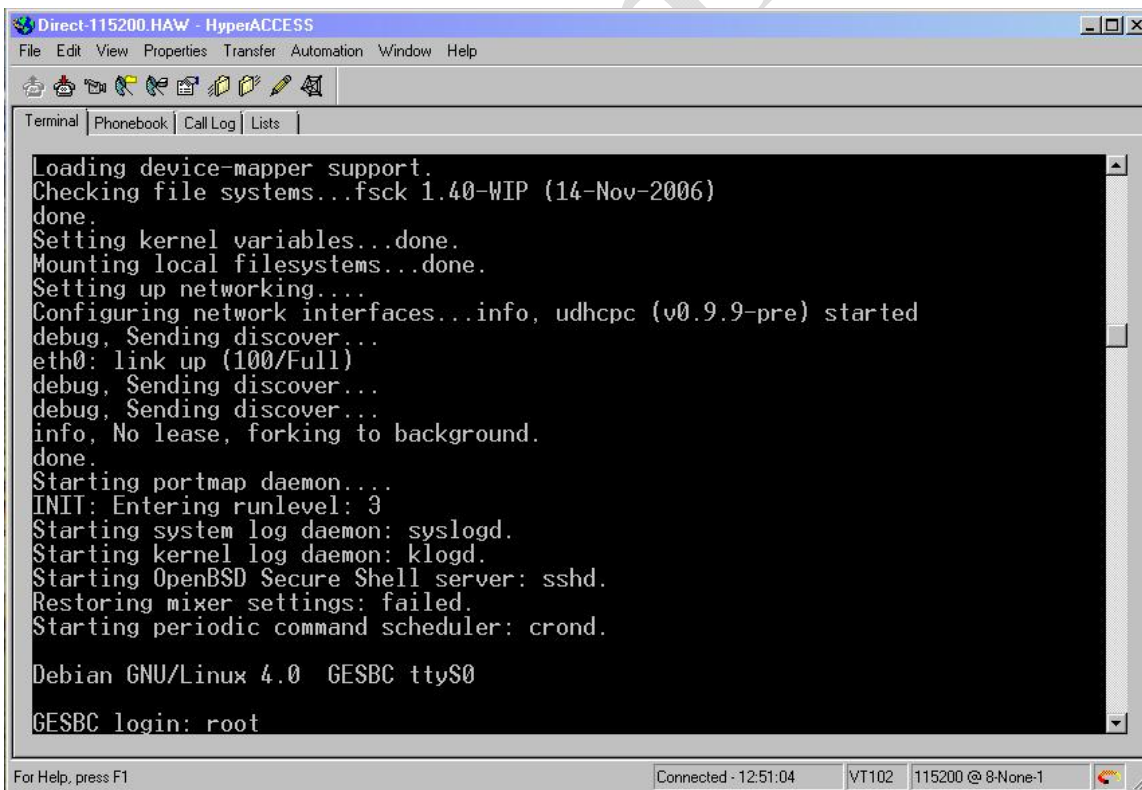
RomBOOT
>

U-Boot 1.3.3 (Jul 19 2008 - 15:50:33)

DRAM: 64 MB
NAND: 128 MiB
DataFlash: AT45DB642
Nb pages: 8192
Page Size: 1056
Size= 8650752 bytes
Logical address: 0xD0000000
Area 0: D0000000 to D00041FF (RO) Bootstrap
Area 1: D0004200 to D00083FF Environment
Area 2: D0008400 to D0041FFF (RO) U-Boot
Area 3: D0042000 to D0251FFF Kernel
Area 4: D0252000 to D083FFFF FS
*** Warning - bad CRC or NAND, using default environment

In: serial
Out: serial
Err: serial
Net: macb0
macb0: Starting autonegotiation...

For Help, press F1
Connected - 12:47:16 VT102 115200 @ 8-None-1
```



The screenshot shows a terminal window titled "Direct-115200.HAW - HyperACCESS". The terminal output displays the Linux boot process. It starts with "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", and "GESBC login: root". The status bar at the bottom indicates "Connected - 12:51:04", "VT102", and "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.

Preliminary

## Chapter 3 – GESBC-9G20i Function Blocks

### AT91SAM9G20

The GESBC-9G20i Single Board Computer uses the Atmel AT91SAM9G20 as the core processor on this development board. The top-level features of AT91SAM9G20 processor are the following:

- ARM926EJ-S RISC Core Processor
- 400 MHz / 400 MIPS Performance
- 32Kbyte Instruction Cache
- 32Kbyte Data Cache
- Linux and Windows CE enabled MMU
- 133 MHz System Bus
- 32 bit SDRAM Interface
- Serial EEPROM Interface
- 10 / 100 Mbps Ethernet MAC
- 6 UART
- Two-port USB Host
- 4 channel 10 bit ADC
- 2 SPI Port
- Serial Audio Interface
- JTAG Interface

More detailed information regarding the AT91SAM9G20 processor can be found at [www.atmel.com](http://www.atmel.com).

### SDRAM

The AT91SAM9G20 features a unified memory address model where all memory devices are accessed over a common address and data bus. The GESBC-9G20i supports 32MB SDRAM.

### FLASH

The GESBC-9G20i is shipped with 128Mbytes of NAND FLASH memory.

### RS-232 Port 0, 1, and 2

The GESBC-9G20i Single Board Computer is shipped with two 3-wire RS-232 UART interface, and one 9 wire RS-232 UART interface.

The port 0 is the debug USART port of the AT91SAM9G20. The P0 connector is the 3 pin header on GESBC-9G20i. The signal designation is listed in the following tables.

**Table 1 UART Port P0 Connector on GESBC-9G20i**

Pin Number	Signal Name
1	RX
2	TX
3	GND

The serial port 1 is the USART 0 on the AT91SAM9G20 processor. It is provided via the DB-9 connector on GESBC-9G20i. The UART P1 provides RTS and CTS hardware handshake signals.

**Table 2 UART Port P1 Connector**

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		

The port 2 is the USART 1 of the AT91SAM9G20. It is provided via the I/O block. Please see Table 4 J8 I/O Expansion for detail.

### RS-485

The GESBC-9G20i Single Board Computer provides one half duplex RS-485 port. The RS-485 port is connected to USART3 with RTS signal for RS-485 driver direction control. The RS-485 signal is provided via the I/O terminal block. Please see Table 4 J8 I/O Expansion for detail. J6 enables the on-board 120 ohm termination resistor. The RTC3 is connected to the RS-485 driver chip for data direction control. The normal setting of RTS signal is normally low. For RS-485 mode the RTS signal must set to normally high. The user program must set the RTS mode before the RS-485 port can be used.

### I2C Bus

The GESBC-9G20i Single Board Computer provides one I2C bus interface via a 1x3 2.54mm spacing header J9.

**Table 3 J9 I2C bus**

Pin Number	Signal Name
1	SDA

2	SCL
3	GND

## Ethernet

The GESBC-9G20i Single Board Computer is shipped with support for a complete Ethernet interface. The AT91SAM9G20 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 AT91SAM9G20 supports 1/10/100 Mbps transfer rates and interfaces to industry standard physical layer devices. The GESBC-9G20i is shipped with the DM9161A 100Base-X / 10Base-T Transceiver device which, along with a RJ45 connector, provides the physical layer interface.

## USB Port

The GESBC-9G20i Single Board Computer provides two USB host connections. The AT91SAM9G20 USB host controller is configured for two root hub ports and features an integrated transceiver for each port. The AT91SAM9G20 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 GESBC-9G20i Single Board Computer provides one USB device port. The USB Device Port (UDP) is compliant with the Universal Serial Bus (USB) V2.0 full-speed device specification.

The GESBC-9G20i Single Board Computer is shipped with one USB host port on standard USB type-A connector and one USB host port on a 6 position 1mm spacing miniature connector that can interface directly to Via VNT6656 USB WiFi module.

The GESBC-9G20i Single Board Computer is shipped with one standard USB type B device port

## I/O Block

The GESBC-9G20i provides 8 protected digital input channels and 6 open collector digital output channels. The digital input channels are protected up to 20V DC and the output channels are capable to sink upto 200mA current. In addition the I/O block also provide one RS-232 port and one RS-485 port. The I/O block uses 20 position 3.5mm spacing terminal block that accepts 18 – 28 gauge wires. The signal assignment is listed in table below.

**Table 4 J8 I/O Expansion**

Pin	I/O Line	Peripheral A	Peripheral B	Comments	Function
1	PC16				Protected digital input 1
2	PC17				Protected digital input 2
3	PC18	SPI1_MISO	TIOA3		Protected digital input 3
4	PC19	SPI1_MOSI	TIOB3		Protected digital input 4
5	PC20	SPI1_SPCK	TIOA4		Protected digital input 5

6	PC21	SPI1_NPCS0	TIOA5		Protected digital input 6
7	PB22	DSR0			Protected digital input 7
8	PB23	DCD0			Protected digital input 8
9	PB12	TXD5	ISI_D10		Open collector output 1
10	PB13	RXD5	ISI_D11		Open collector output 2
11	PB16	TK0	TCLK3		Open collector output 3
12	PB17	TF0	TCLK4		Open collector output 4
13	PB18	TD0	TIOB4		Open collector output 5
14	PB19	RD0	TIOB5		Open collector output 6
15					GND
16	RX2			USART2	RS-232 port 2 receiving line
17	TX2			USART2	RS-232 Port 2 transmitting line
18	RX3			USART3	RS-485 port receiving line
19	TX3			USART3	RS-485 port transmitting line
20					GND

#### 4 x 20 Character LCD Port J30

The GESBC-9G20i provides a LCD port for Hitachi HD44780 compatible character LCD displays. The LCD port signal assignment is listed in the following table.

**Table 5 J30 LCD**

Pin Number	Signal Name	Comment
1	VSS	
2	VDD	
3	VO	
4	RS	
5	R/W	
6	E	
7	DB0	
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	BLA	Controlled by PC31
16	BLK	

#### RTC

The GESBC-9G20i uses the AT91SAM9G20 on-chip RTC with battery hook-up to provide accurate time keeping. The on-board battery holder accepts CR1225/CR1220 coin cell batteries.

## ZigBee Interface

The GESBC-9G20i Single Board Computer is shipped with socket for XBeeZnet2.5 module from Digi. The serial data lines connected to the ZigBee interface are RXD1 and TXD1 from the AT91SAM9G20 processor. The hardware control signals are RTS1 (PB28) and CTS1 (PB29) from the AT91SAM9G20 processor. The system reset signal NRST is connected to the module RESET line pin 5.

## Optional A/D

The GESBC-9G20i Single Board Computer provides support for optional 12 bit 8 channel A/D. 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 ksp/s. The 8 single ended analog inputs can be also configured as 4 pairs of differential input channels. The optional A/D interface is provided via a 2x8 2.54mm header. The signal designation is listed in the following table.

**Table 6 J12 Optional A/D**

Pin Number	Signal Name	Pin Number	Signal Name
1	AIN0	2	AIN1
3	AIN2	4	AIN3
5	AIN4	6	AIN5
7	AIN6	8	AIN7
9	AGND	10	AGND
11	DIO0	12	DIO1
13	DIO2	14	DIO3
15	GND	16	GND

## JTAG

The GESBC-9G20i 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 7 J20 JTAG Connector**

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

## Power Requirement

The GESBC-9G20i Single Board Computer has on-board voltage regulator that accepts 7.5 ~ 24V DC power input.

**Table 8 J1 Power Supply Connector**

Pin Number	Signal Name
1	+7.5V ~ +24V DC
2	GND

Preliminary

## Chapter 4 – Software Description

### Overview

This chapter provides information regarding the software that is shipped with the GESBC-9G20i Board. The software included with the board is U-boot boot loader, Linux kernel 2.6.25, and Debian distribution style compact root file system. The applications included provide access to all hardware functions on the GESBC-9G20i board.

### Data Storage on GESBC-9G20i

The default configuration of the GESBC-9G20i 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 factory default storage map on the NAND FLASH.

**Table 9 NAND FLASH Storage Map**

Start Address	Size	Usage
0x00000000	0x20000	Boot strap code
0x00020000	0x40000	U-boot
0x00060000	0x20000	U-boot primary environment storage range
0x00080000	0x20000	U-boot secondary environment storage range
0x00100000	0x300000	Linux kernel
0x00400000	--	Root file system

### GESBC-9G20i Linux Code

The GESBC-9G20i is shipped with Linux 2.6.27 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-9G20i board.

### U-boot

U-boot provides a simple interface for loading operating systems and applications onto the GESBC-9G20i 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.

```
U-Boot> printenv
bootargs=console=ttyS0,115200 root=/dev/mtdblock2 rw rootfstype
=jffs2 mtdparts=atmel_nand:1M(bootloader),3M(kernel),-(rootfs)
bootcmd=nand read.jffs2 0x22000000 0x100000 0x200000; bootm
bootdelay=1
baudrate=115200
ethaddr=00:0c:20:02:0a:5b
ipaddr=192.168.1.200
serverip=192.168.1.199
netmask=255.255.255.0
stdin=serial
stdout=serial
stderr=serial
ethact=macb0

Environment size: 353/131067 bytes
U-Boot>
```

The `bootcmd` setting of the U-boot reads the Linux kernel from NAND FLASH at address 0x100000 to SDRAM at address 0x22000000 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-9G20i 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-9G20i board.

```
RomBOOT
>
U-Boot 1.3.4 (Jul 19 2009 - 15:50:33)

DRAM: 32 MB
NAND: 128 MiB
In: serial
Out: serial
Err: serial
Net: macb0
macb0: Starting autonegotiation...
macb0: Autonegotiation timed out (status=0x7849)
macb0: link up, 100Mbps full-duplex (lpa: 0x4de1)
Hit any key to stop autoboot: 0
U-Boot>
```

The network 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 0x21000000 uImage
```

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

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 following command to store the kernel image from SDRAM to NAND FLASH,

```
nand write.jffs2 0x21000000 0x100000 0x200000
```

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

```
nand erase 0x400000 [available_nand_flash_memory_size]
t 0x21000000 rootfs.img
nand write.jffs2 0x21000000 0x400000 $(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 AT91SAM9G20 System-on-a-Chip processor. The central processing core on the AT91SAM9G20 is a 400 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-9G20i 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-9G20i is shipped with Linux kernel version 2.6.27. The patch for the kernel source tree is available at <http://www.linux4sam.org/twiki/bin/view/Linux4SAM/LinuxKernel>.

#### *Prepare Linux Kernel source*

Obtain the kernel source 2.6.27 from <http://www.kernel.org>. Untar the Linux kernel,

```
tar xjf linue-2.6.27.bz2
```

Patch the kernel source with patches for Atmel AT91SAM9G20-EK,

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

#### *Configure Linux Kernel*

The GESBC-9G20i can use the default configuration file for the Atmel AT91SAM9G20-ek evaluation board.

```
make ARCH=arm CROSS_COMPILE=arm-unknown-linux-gnu-  
AT91SAM9G20ek_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<sup>1</sup>,

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

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

---

<sup>1</sup> The U-boot tool `mkimage` must be pre-installed in order to make final `uImage`.

## 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.

Preliminary