GLOMATION

Embedded Single Board Computer

GESBC-9G20u User’s Manual

Table of Contents

List of Tables

Chapter 1 – Introducing the GESBC-9G20u Single Board Computer

GESBC-9G20u Overview

The GESBC-9G20u 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-9G20u is well suited for industrial controls, set-top boxes, point-of-sale terminals, biometric security systems, and building automation devices.

Advanced Features

The heart of the GESBC-9G20u 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 ideal platform for battery operated applications.

The list below summarizes the features of the GESBC-9G20.

- 400MHz 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
- 1 RS-232 Universal Asynchronous Receiver / Transmitters (UARTs)
- 2 USB Host Port
- 1 USB Device Port
- Real-Time Clock with battery backup connector
- Hardware Debug Interface
- Micro SD/MMC Socket
- 1 I2C Port
- 1 SPI Port
- Up to 36 GPIO

Figure 1 below shows a picture of the GESBC-9G20u Single Board Computer.

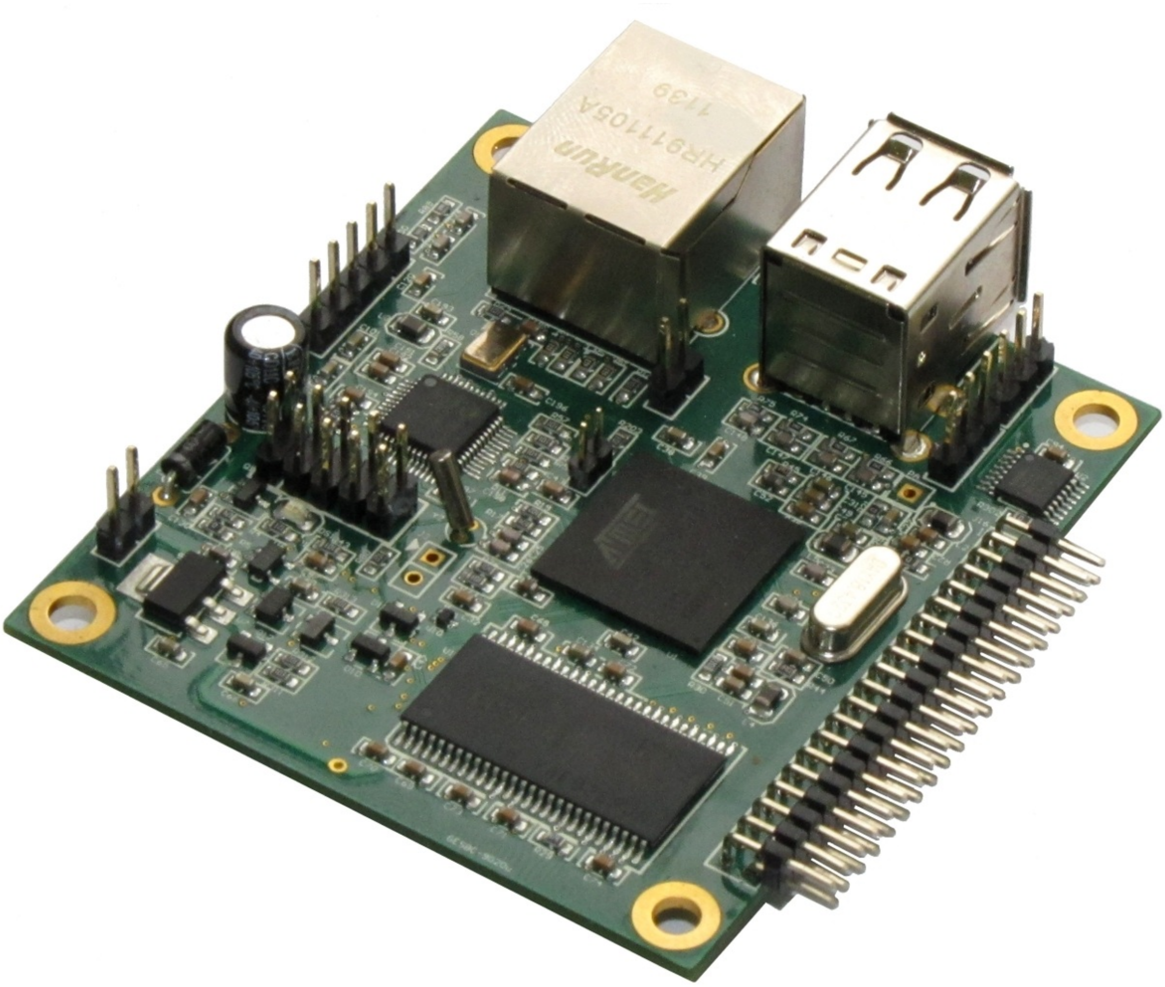


Figure 1. GESBC-9G20u Single Board Computer

Chapter 2 – Getting Started

This chapter describes the GESBC-9G20u 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-9G20u Single Board Computer

- Operation
  - Describes how to operate the GESBC-9G20u Single Board Computer

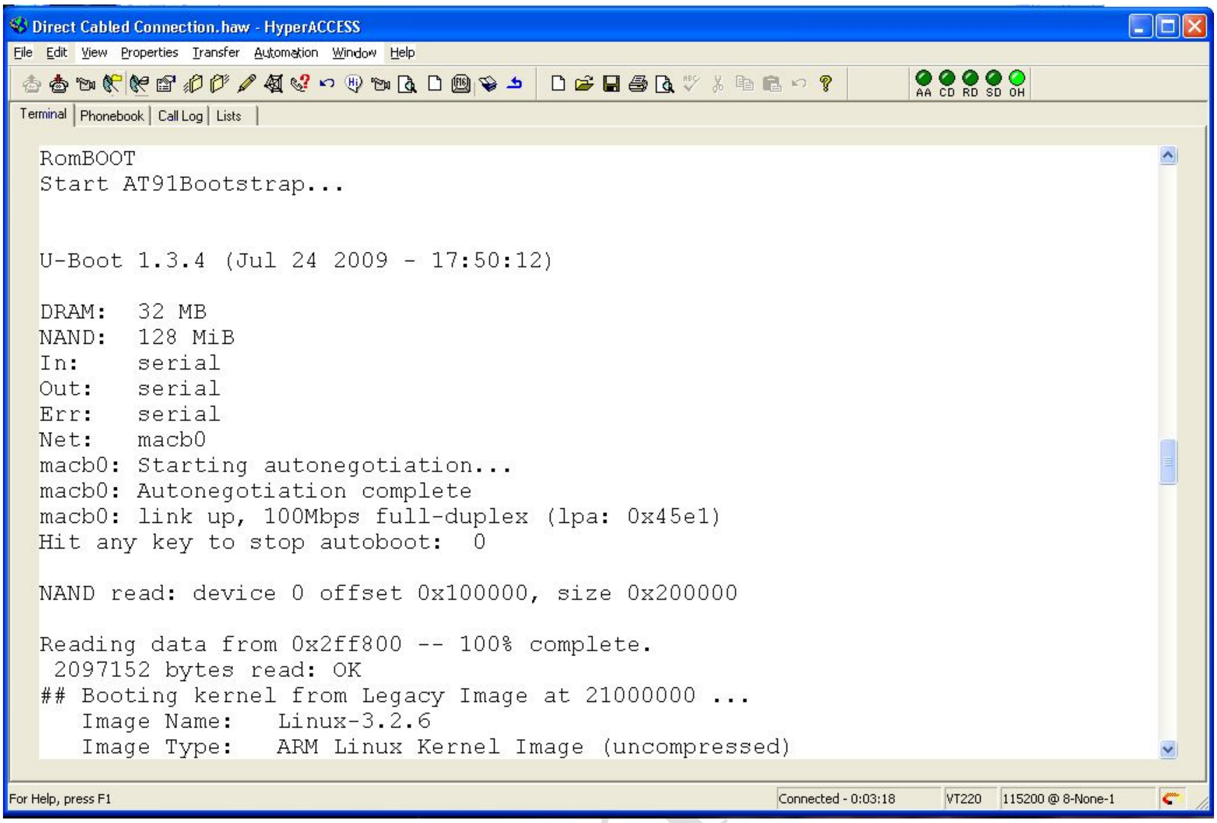
Assembly and Connections

In order to use the GESBC-9G20u the user must first assemble and connect the peripherals to the GESBC-9G20, as described in the following procedure.

1. Place the GESBC-9G20u on a static free surface.
2. Connect 5V regulated power supply to the board.
3. Connect null modem serial cable between the GESBC-9G20u debug portP0 and the PC/terminal serial port.
4. Launch a terminal emulator, such as HyperTerminal, or minicom, on the PC configured to connect to the serial port of the GESBC-9G20. Configure the serial port with the following parameters: 115200 bits per second, 8 data bits, no parity, 1 stop bit, no flow control.
5. Connect the board to a local area network (optional)

Operation

A few seconds after applying power to the GESBC-9G20, 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
AA CD RD SD OH
Terminal | Phonebook | Call Log | Lists
For Help, press F1
Connected - 0:03:18
VT220
115200 @ 8-None-1

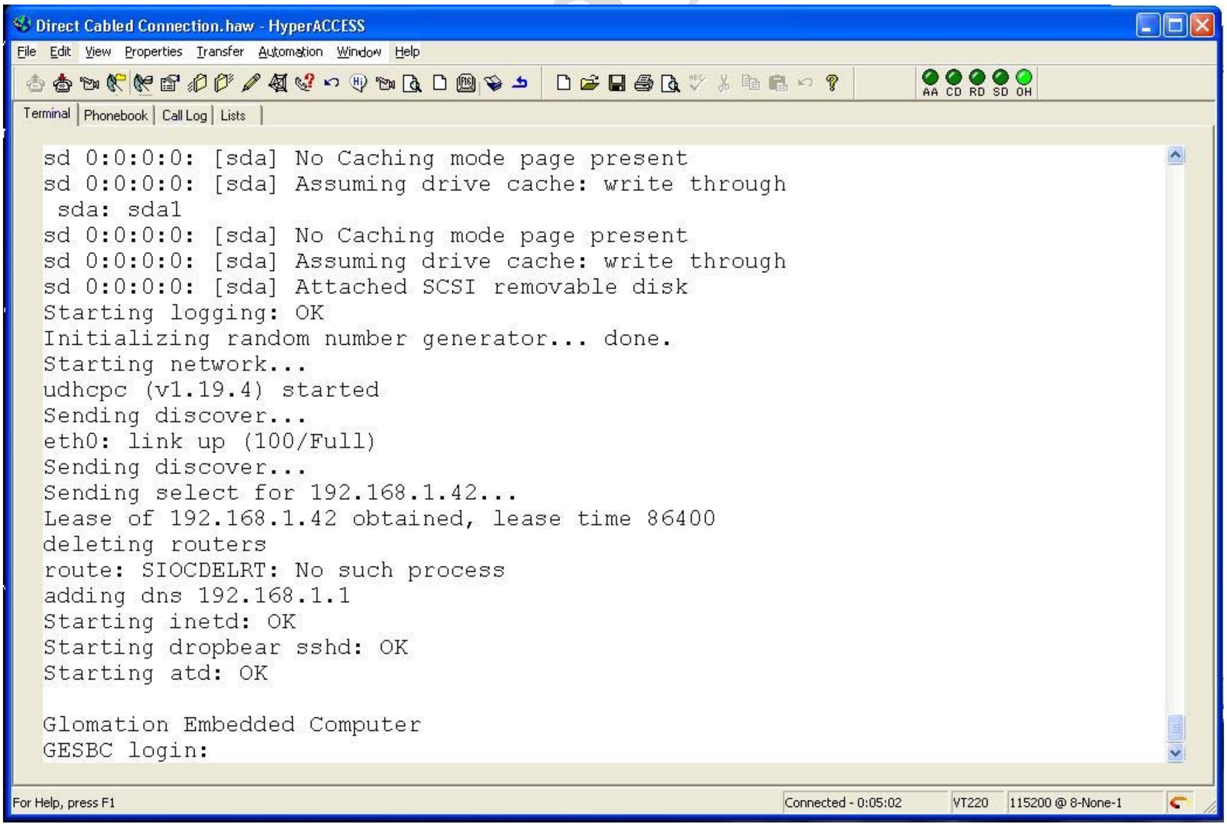
```
RomBOOT
Start AT91Bootstrap...


U-Boot 1.3.4 (Jul 24 2009 - 17:50:12)

DRAM:  32 MB
NAND:  128 MiB
In:    serial
Out:   serial
Err:   serial
Net:   macb0
macb0: Starting autonegotiation...
macb0: Autonegotiation complete
macb0: link up, 100Mbps full-duplex (lpa: 0x45e1)
Hit any key to stop autoboot:  0

NAND read: device 0 offset 0x100000, size 0x200000

Reading data from 0x2ff800 -- 100% complete.
 2097152 bytes read: OK
## Booting kernel from Legacy Image at 21000000 ...
   Image Name:   Linux-3.2.6
   Image Type:   ARM Linux Kernel Image (uncompressed)
```


Direct Cabled Connection.haw - HyperACCESS
File Edit View Properties Transfer Automation Window Help
AA CD RD SD OH
Terminal | Phonebook | Call Log | Lists
For Help, press F1
Connected - 0:05:02
VT220
115200 @ 8-None-1

```
sd 0:0:0:0: [sda] No Caching mode page present
sd 0:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1
sd 0:0:0:0: [sda] No Caching mode page present
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Attached SCSI removable disk
Starting logging: OK
Initializing random number generator... done.
Starting network...
udhcpc (v1.19.4) started
Sending discover...
eth0: link up (100/Full)
Sending discover...
Sending select for 192.168.1.42...
Lease of 192.168.1.42 obtained, lease time 86400
deleting routers
route: SIOCDELRT: No such process
adding dns 192.168.1.1
Starting inetd: OK
Starting dropbear sshd: OK
Starting atd: OK

Glomation Embedded Computer
GESBC login:
```

Configurations

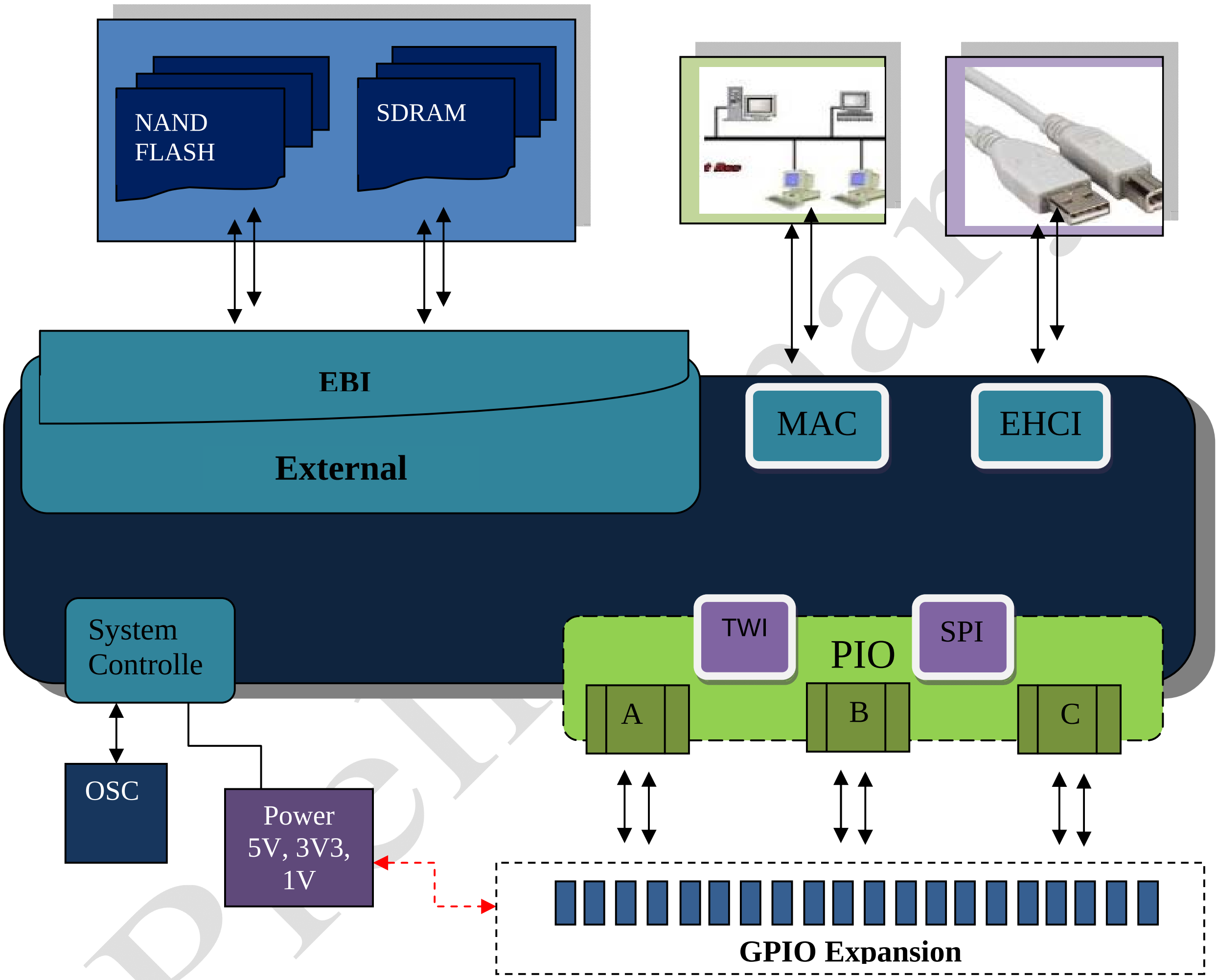
Jumpers are used to configure the GESBC-9G20u to operate in different mode. The following table lists all the settings for each jumper.

Table 1 System Configuration

Jumper	Description
S1	PA31 port input for boot strap code boot mode open – normal ROM boot sequence close – Bootstrap code erases first FLASH memory sector, system reverts to virgin state
BP1	System reset

Chapter 3 – GESBC-9G20u Function Blocks

The figure below shows the basic system functional blocks of the GESBC-9G20u.



AT91SAM9G20

The GESBC-9G20u 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
- 32 bit SRAM / FLASH / ROM 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.

SDRAM

The AT91SAM9G20 features a unified memory address model where all memory devices are accessed over a common address and data bus. The GESBC-9G20u is equipped with 32MB SDRAM.

FLASH

The GESBC-9G20u is shipped with 128 Mbytes of NAND FLASH memory. The GESBC-9G20u can be also ordered with optional 512MB ~ 1GB NAND FLASH.

RS-232 Port 0 (Debug port)

The GESBC-9G20u Single Board Computer is shipped with one 3-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-9G20. The signal designation is listed in the following tables.

Table 2 UART Port P0 Connector on GESBC-9G20

Pin Number	Signal Name

1	RX
2	TX
3	GND

Ethernet

The GESBC-9G20u 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-9G20u 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-9G20u Single Board Computer is shipped with 2 USB host port on standard USB type-A double deck connector.

The GESBC-9G20u Single Board Computer is shipped with one USB device port J17. The USB device port signal assignment is listed in the following table.

Table 3 J17 USB Device Port

Pin Number	Signal Name
1	USB connection
2	DM
3	DP
4	GROUND
5	GROUND

SPI Bus, I2C, On-chip A/D and GPIO

The AT91SAM9G20 contains very rich set of peripherals that are multiplex into 2 groups, Peripheral A and Peripheral B, with individually programmable pins. The SPI bus, A/D, I2C, and GPIO are provided together with other functions on the I/O expansion port. The I/O expansion port is a 2x20 2mm spacing header. The following table lists signals available on the I/O expansion connector with their corresponding multiplexed functions and default usage on the GESBC-9G20u Single Board Computer.

Table 4 J16 I/O Expansion

Pin	I/O Line	Peripheral A	Peripheral B	Comments	Function
1					+3.3V or +5V

2					+3.3V or +5V
3	PB0	SPI1_MISO	TIOA3		
4	PB1	SPI1_MOSI	TIOB3		
5	PB2	SPI1_SPCK	TIOA4		
6	PB3	SPI1_NPCS0	TIOA5		
7	PB4	TXD0			
8	PB5	RXD0			
9	PB6	TXD1	TCLK1		
10	PB7	RXD1	TCKL2		
11	PB8	TXD2			
12	PB9	RXD2			
13	PB10	TXD3	ISI_D8		
14	PB11	RXD3	ISI_D9		
15	PB12	TXD5	ISI_D10		
16	PB13	RXD5	ISI_D11		
17	PA24	SCL			
18	PA23	SDA			
19	PB16	TK0	TCLK3		
20	PB17	TF0	TCLK4		
21	PB18	TD0	TIOB4		
22	PB19	RD0	TIOB5		
23	PB20	RK0	ISI_D0		
24	PB21	RF0	ISI_D1		
25	PB22	DSR0	ISI_D2		
26	PB23	DCD0	ISI_D3		
27	PB24	RTR0	ISI_D4		
28	PB25	RI0	ISI_D5		
29	PB26	RTC0	ISI_D6		
30	PB27	CTS0	ISI_D7		
31	PB28	RTS1	ISI_PCK		
32	PB29	CTS1	ISI_VSYNC		
33	PB30	PCK0	ISI_HSYNC		
34	PB31	PCK1	ISIMCK		
35	PC0	AD0	SCK3		
36	PC1	AD1	PCK0		
37	PC2	AD2	PCK1		
38	PC3	AD3	SPI1_NPCS3		
39					GND
40					GND

For more detailed information on multiplexed peripherals please see AT91SAM9G20 data sheet.

The pin 1 and pin 2 power source can be selected with a jumper to set it to 3.3V, 5V or none. The voltage select jumper is J18.

Table 5 J18 GPIO Header Voltage Select

Pin Number	Signal Name
1	+3.3V
2	Connection to Pin 1 and pin 2 of J16
3	+5V

RTC

The GESBC-9G20u uses the AT91SAM9G20 on-chip RTC with battery hook-up to provide accurate time keeping. The RTC back battery can be connected via J3.

Table 6 J3 RTC Backup Battery Connection

Pin Number	Signal Name
1	VBat
2	Ground

JTAG

The GESBC-9G20u 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-9G20u Single Board Computer requires regulated 5V DC. The power supply should have minimum 300mA capacity.

Table 8 J1 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-9G20u 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-9G20u board.

Data Storage on GESBC-9G20u

The default configuration of the GESBC-9G20u 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-9G20u Linux Code

The GESBC-9G20u is shipped with Linux 3.2 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-9G20u board.

U-Boot

U-boot provides a simple interface for loading operating systems and applications onto the GESBC-9G20u 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 0x21000000 0x100000 0x200000; bootm
bootdelay=1
baudrate=115200
ethaddr=00:0c:20:02:0a:5b
ipaddr=192.168.0.200
serverip=192.168.0.102
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 0x21000000 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-9G20u 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-9G20u board.

```
RomBOOT
>

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

DRAM:  64 MB
NAND:  256 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 flowing 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 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 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 VMWare virtual machine image with cross development tools pre-installed is available at Glomation website http://www.glomationinc.com/support.html. User can download VMPlayer from VMWare website to run the virtual machine image. This guide assumes user had basic Linux or Unix application development knowledge.

Compile Linux Kernel

The GESBC-9G20u is shipped with Linux kernel version 3.2.6. The main line Linux kernel includes all the driver for peripherals on the GESBC-9G20u.

Prepare Linux Kernel source

Obtain the kernel source 3.2.6 from http://www.kernel.org. Untar the Linux kernel,

```
tar xjf  linue-3.2.6.bz2
```

Configure Linux Kernel

The GESBC-9G20u 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[1],

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

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

[1] 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.

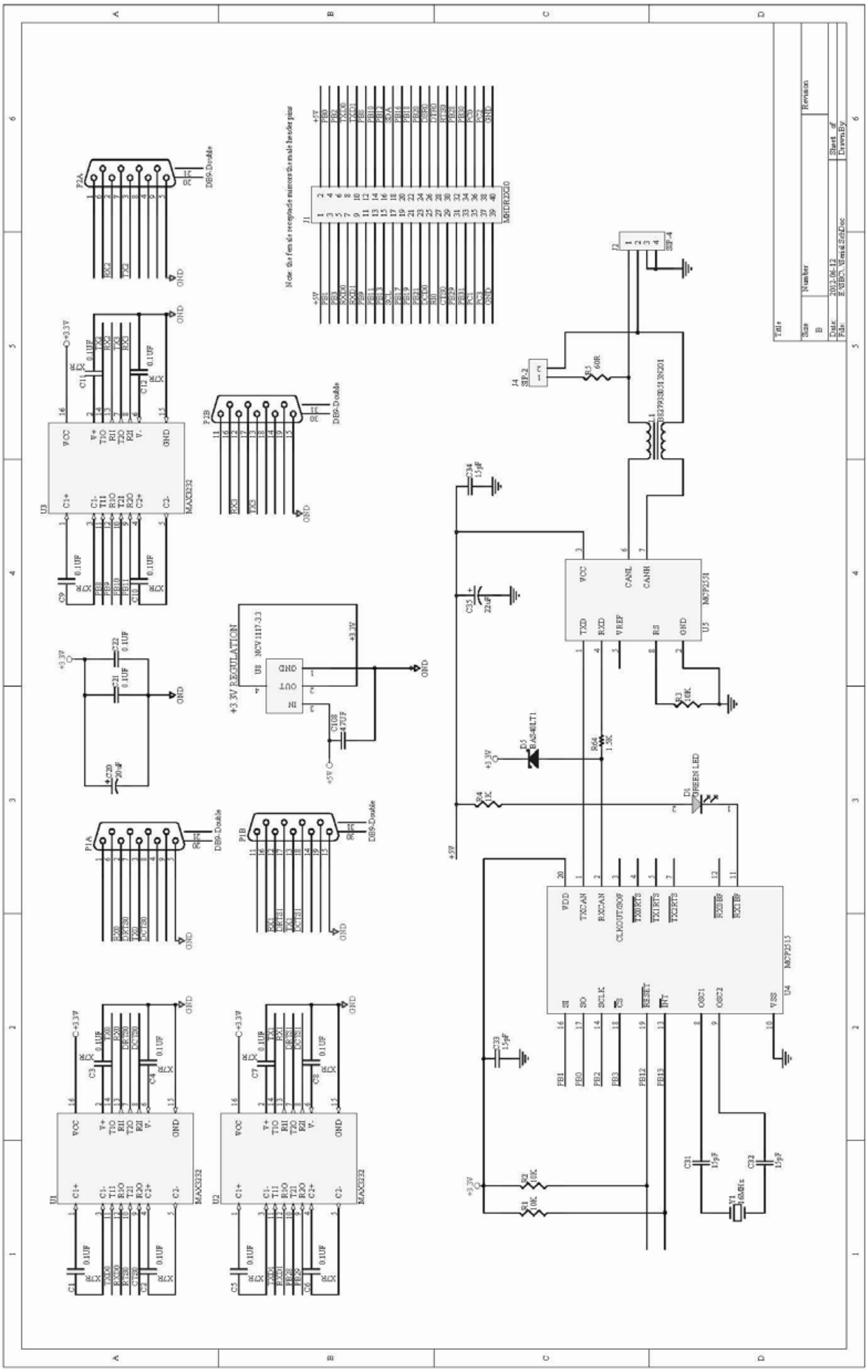
Appendix

RS-232 and CAN Bus Expansion Card

The GPIO expansion port on the GESBC-9G20u consists of 36 configurable signal line. Each line can be configured as general purpose I/O or many other peripheral functions provided by the versatile AT91SAM9G20 processor core. The following example shows a simple peripheral expansion board expands the GESBC-9G20u with 4 RS-232 port (2 with hardware handshake) and 1 CAN bus port.

The CAN bus transceiver MCP2515 connects to the SPI bus of the GESBC-9G20u through the GPIO header J16 on the GESBC-9G20u. GPIO line PB12 and PB 13 are used as the reset control of the MCP-2512 and interrupt request. The MCP2515 connects to the 3.3V power rail to ease the interface to the 3.3V AT91SAM9G20 GPIO logic circuit. The CAN bus signal driver MCP2551 operates at 5V power rail to drive the high voltage CAN bus signal. A diode clamps the MCP2551 output to 3.3V for the CAN bus transceiver MCP2515.

Three RS-232 driver chips are used to facilitate four RS-232 serial port. RS-232 port 1 and RS-232 port 2 have RTC and CTS hardware flow control signals. The RS-232 port 3 and RS-232 port 4 are 3-wire RS-232 ports.


+3.3V REGULATION
U8 NCV1117-3.3
IN
OUT
GND
C108 47UF
+5V
+3.3V
GND
C20 20uF
C21 0.1UF
C22 0.1UF
U1 MAX3232
U2 MAX3232
U3 MAX3232
VCC
V+
V-
C1+
C1-
C2+
C2-
T1O
T2O
T1I
T2I
R1I
R2I
R1O
R2O
GND
0.1UF
P1A DB9-Double
P1B DB9-Double
P2A DB9-Double
P2B DB9-Double
J1 MHDR2X20
Note: the female receptacle mirrors the male header pins
U4 MCP2515
VDD
TXCAN
RXCAN
CLKOUT/SOF
TX0RTS
TX1RTS
TX2RTS
RX0BF
RX1BF
SI
SO
SCLK
CS
RESET
INT
OSC1
OSC2
VSS
U5 MCP2551
TXD
RXD
VREF
RS
CANL
CANH
R1 10K
R2 10K
R3 10K
R4 1K
R5 60R
R64 1.5K
D1 GREEN LED
D5 BAS40LT1
Y1 16MHz
C31 15pF
C32 15pF
C33 15pF
C34 15pF
C35 22uF
B82793S0513N201
J2 SIP-4
J4 SIP-2
PB0
PB1
PB2
PB3
PB12
PB13
Title
Size
B
Number
Revision
Date 2012-06-12
Sheet of
File
Drawn By
A
B
C
D
1
2
3
4
5
6