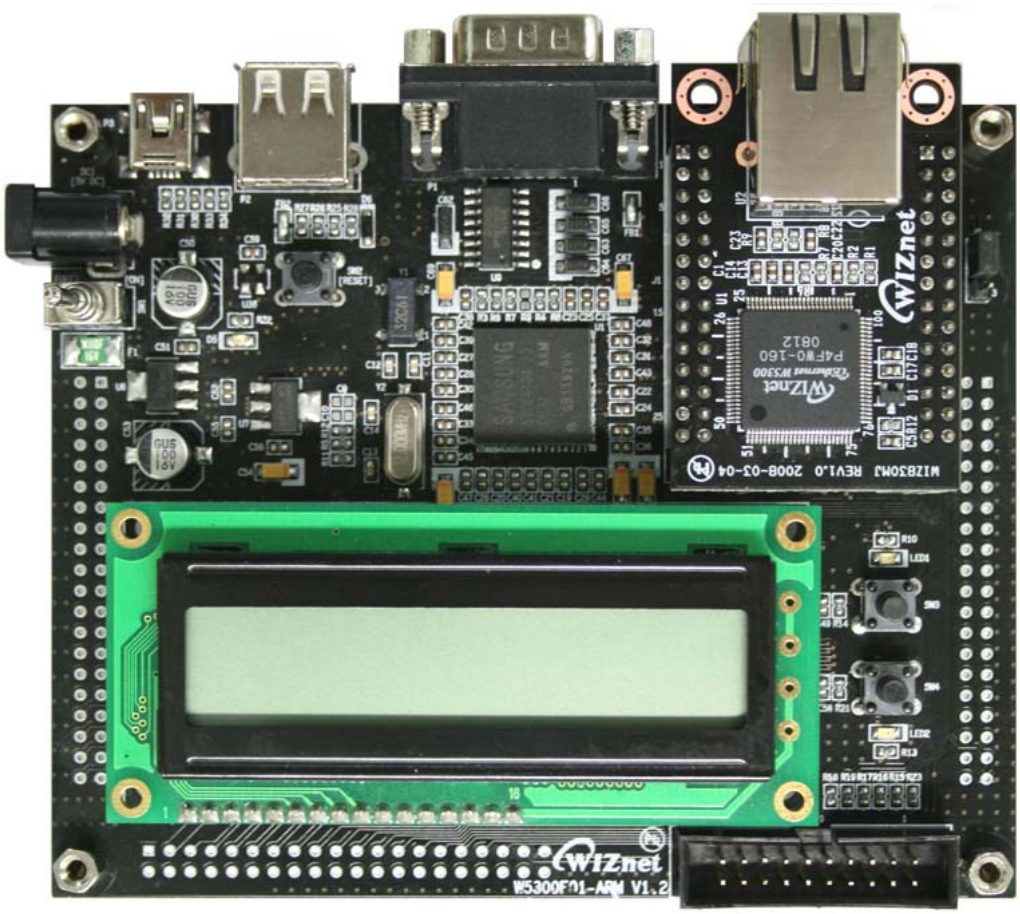
W5300E01-ARM User's Manual

(Version 1.0)



☞ For more information, visit our website at http://www.wiznet.co.kr

Document History Information

Revision	Data	Description
Ver. 1.0	July 15, 2008	1st Release

WIZnet's Online Technical Support

If you have something to ask about WIZnet Products, Write down your question on Q&A Board in WIZnet website (www.wiznet.co.kr). WIZnet Engineer will give an answer as soon as possible.

COPYRIGHT NOTICE

Technical Support: support@wiznet.co.kr
Sales & Distribution: sales@wiznet.co.kr

For more information, visit our website at http://www.wiznet.co.kr

Table of Contents

Figures

Tables

1. Overview

W5300E01-ARM is the test board to evaluate the function of W5300 based on ARM920T.

1.1. Products Introduction

W5300E01-ARM is composed of the base board and WZ830MJ module. In the base board, ARM920T based Samsung S3C2410A-200MHz processor is built in, and W5300, the hardware TCP/IP chip is used.

By using serial port of base board and Ethernet port of WIZ830MJ, the communication environment can be easily set up. You can also test external devices having USB interface through USB Host and Device port.

W5300E01-ARM operates on Linux OS. By installing 64MB SDRAM, various applications can operate without any problem.

Samsung S3C2410A processor supports NAND Flash Booting. By installing 64MB NAND Flash memory, enough space is provided for bootloader, OS and user application.

By using extension connector (40pin * 3, total 120pin), the easy extension is supported for the functions that the base board does not support.

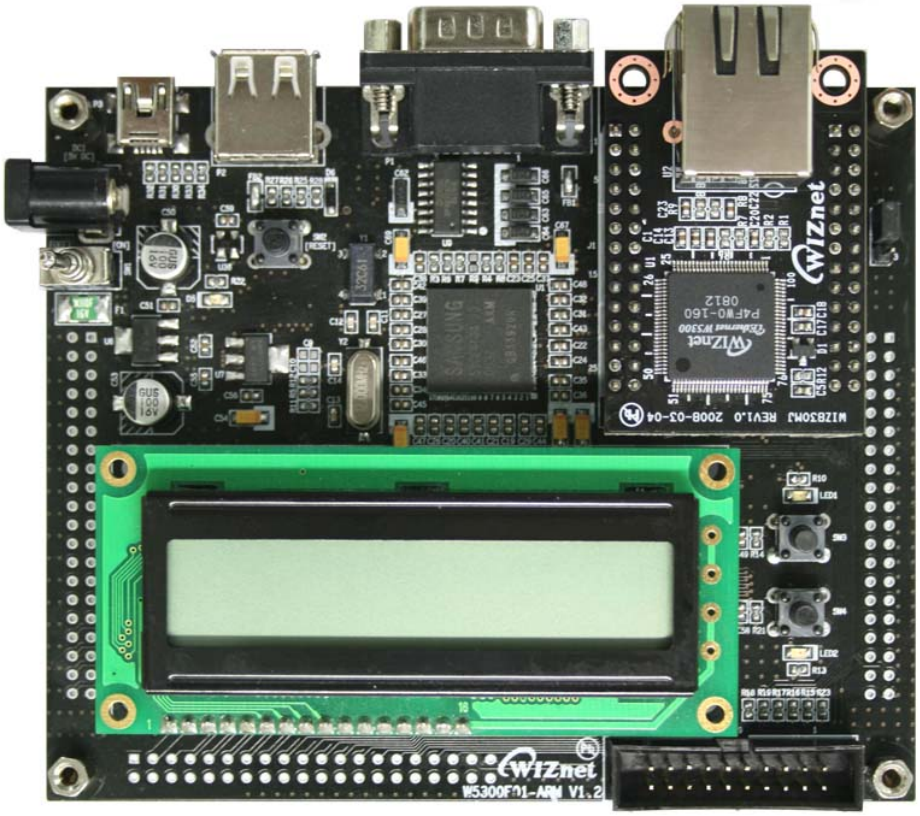


Figure 1-1 : W5300E01-ARM Board

1.2. Products Specification

1.2.1. Base Board Specification

ITEM	Description	ETC
MCU	200MHz Samsung S3C2410A ARM RISC Processor	ARM920T
RAM	SDRAM 64MB	
ROM	NAND Flash ROM 64MB	
Serial	RS-232C 1Port	
USB Host	USB Host 1Port	
USB Device	USB Device 1Port	
Ethernet	Supported by WIZ830MJ Module	Basic ITEM
LCD	16Character * 2Line Character LCD Port	Basic ITEM
LED	LED 2Ea for Debugging	
Button	Tact Switch 2Ea for Debugging	
JTAG	On board JTAG Socket	
WIZ830MJ Module Connector	56Pin (28Pin * 2) 2.54mm Pitch Pin-Header Socket	
Expansion Port	120Pin (40pin * 3) 2.54mm Pitch Pin-Header	
Power	DC 5V / 2A Adapter	Basic ITEM
PCB	118mm * 97mm Size	

Table 1-1 : W5300E01-ARM Base Board Specification

1.2.2. WIZ830MJ Module Specification

ITEM	Description	ETC
Ethernet Chip	WIZnet W5300 TCP/IP Chip	
RJ-45	RJ-45 1Port (integrated Transformer)	
Base board Interface	56Pin (28pin * 2) 2.54mm Pitch Pin-Header	
PCB	50mm * 34mm Size	

Table 1-2 : WIZ830MJ Module Specification

2. Products Description

2.1. Board Layout

2.1.1. Base Board Layout

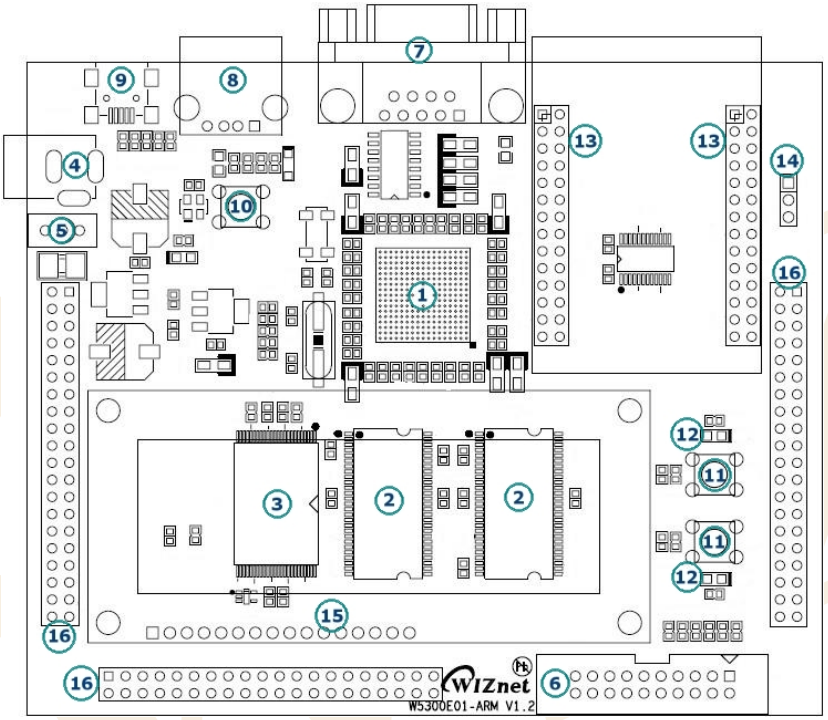


Figure 2-1 : W5300E01-ARM Base Board Layout

2.1.2. WIZ830MJ Module Layout

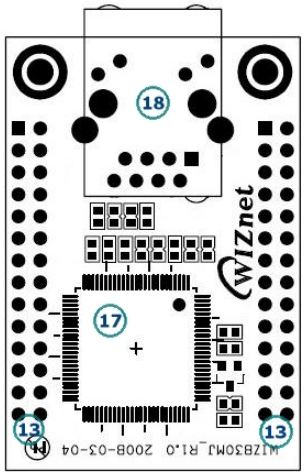


Figure 2-2 : WIZ830MJ Module Layout

2.1.3. Parts Description

The description of each part shown in <Figure 2-1 : W5300E01-ARM Base Board Layout> and <Figure 2-2 : WIZ830MJ Module Layout> is as below.

No	Description	No	Description
1	Samsung S3C2410A Processor	10	Reset Switch
2	32MB SDRAM * 2Ea (Total 64MB)	11	Tact Switch for Debugging * 2Ea
3	64MB NAND Flash ROM (K9F1208)	12	Green LED for Debugging * 2Ea
4	DC 5V / 2A Adapter Jack	13	WIZ830MJ Module Interface Connector
5	Power Switch	14	WIZ830MJ Module Bus width select Jumper
6	JTAG Connector	15	Character LCD Interface Connector
7	RS-232C Serial Connector	16	Expansion Connector (40Pin * 3Ea)
8	USB Host Connector	17	WIZnet W5300 TCP/IP Chip
9	USB Device(Slave) Connector	18	RJ-45 Jack (integrated Transformer)

Table 2-1 : Parts Description of W5300E01-ARM

- For more detail, refer to 3. Hardware Designer's Guide.

2.2. Package and Contents

The contents of W5300E01-ARM are as below.

	Item	Quantity
Board	W5300E01-ARM Base Board	1
	WIZ830MJ Module (plugged in the base board of W5300E01-ARM)	1
	Character LCD (installed in the base board of W5300E01-ARM)	1
Accessory	Data CD	1
	Power adapter (DC 5V / 2A)	1
	UTP Cable	1
	Serial Cable	1
	USB Host / Device Cable	Option

Table 2-2 : W5300E01-ARM Contents

Directory			Contents
W5300E01-ARM	Documents	Manual	User's Manual
		Datasheet	Datasheet of Main Parts
	Hardware	Schematics	W5300E01-ARM Hardware Schematic
		Parts List	W5300E01-ARM Parts List
	Software	Bootloader	wiz-u-boot Source
		LinuxKernel	Linux kernel Source Linux Kernel Patch file Linux Kernel Config file
		Image	Bootloader Image Kernel Image Ramdisk Image
		Tools	Toolchain(compiler, etc…)
		Drivers	W5300 Driver Character LCD Driver
		Examples	Loopback test

Table 2-3 : Contents of Data CD

3. Hardware Designer's Guide

3.1. Block Diagram

3.1.1. System Block Diagram

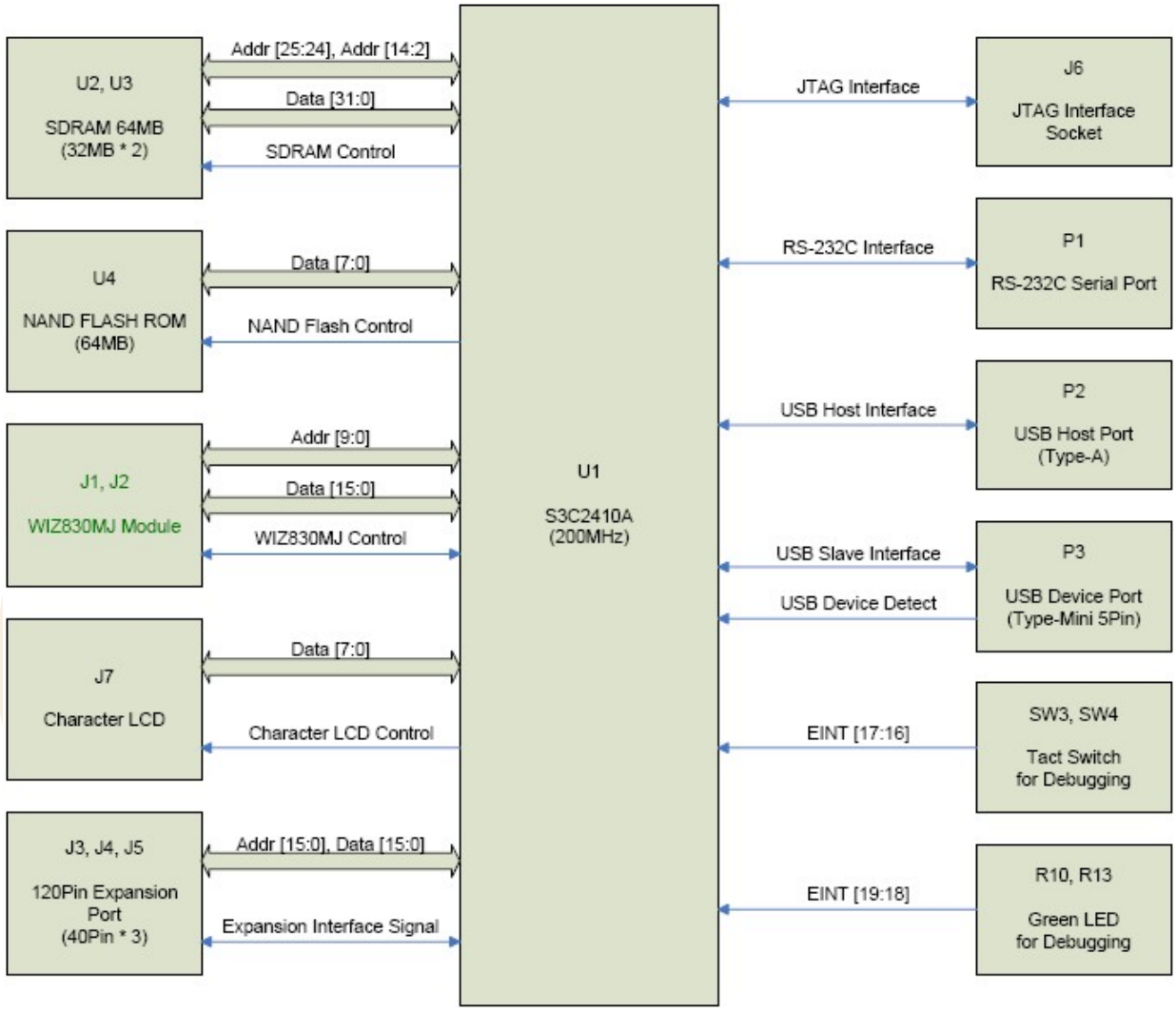


Figure 3-1 : W5300E01-ARM System Block Diagram

3.1.2. Power Block Diagram

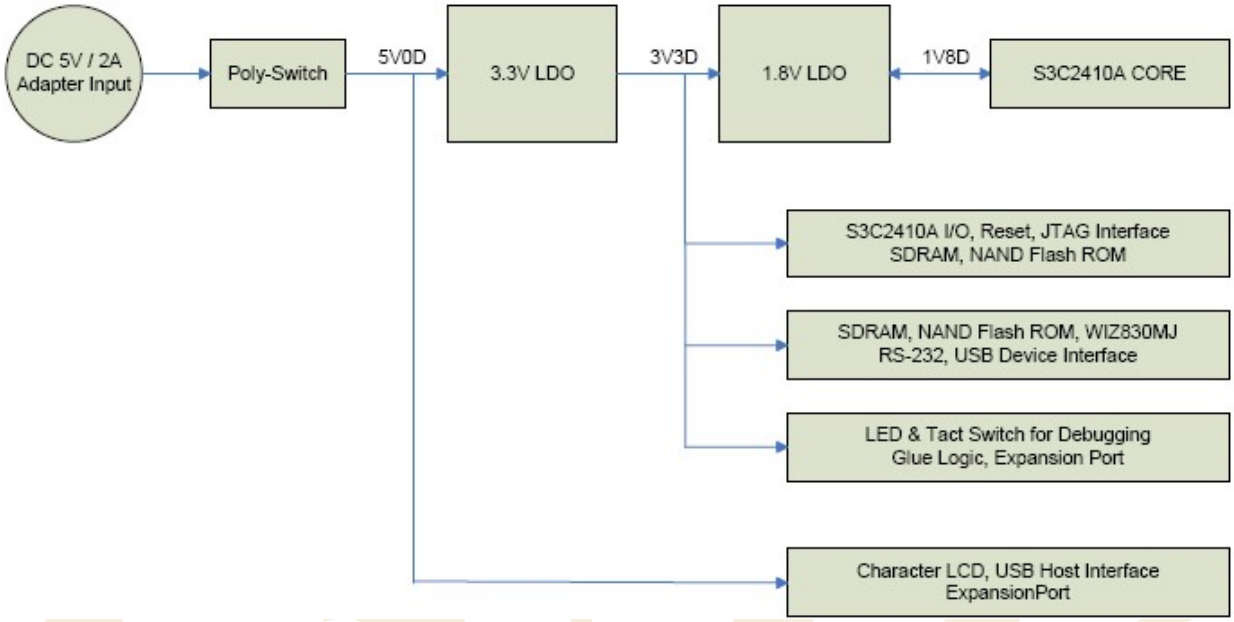


Figure 3-2 : W5300E01-ARM Power Block Diagram

3.2. Block Description

W5300E01-ARM can be divided into below blocks.

- S3C2410A System Block
- SDRAM
- NAND Flash ROM
- WIZ830MJ Module
- Character LCD
- RS-232C Serial Port
- USB Host / Device Port
- JTAG Interface
- LED & Tact Switch for Debugging
- Expansion Port Interface
- Power Block

3.2.1. S3C2410A System Block

In order to support NAND Flash boot loader, Samsung S3C2410A processor contains SRAM buffer called as Steppingstone.

W5300E01-ARM Platform uses NAND Flash ROM for Booting memory. It is designed to be initialized with NAND Flash booting by pull-down OM0 and OM1 pins of S3C2410A to GND.

As power-on reset IC is installed for user manual reset, it is possible to manually reset the board by using tact switch during board operation.

3.2.2. SDRAM

64MByte SDRAM is used for external memory of S3C2410A processor, and provides enough space for operation of O/S and User application.

3.2.3. NAND Flash ROM

64MByte NAND Flash ROM is used for external programming memory of S3C2410A and non-volatile storage device. Basically, Linux bootloader, Kernel, and File System are programmed in this Flash ROM. Additionally, embedded web server (utilizing W5300 TCP/IP) is also saved in NAND Flash ROM. Extra space can be used for user data field.

.

3.2.4. WIZ830MJ Module

WIZ830MJ is the Ethernet module having W5300 TCP/IP chip and RJ-45 connector (having Transformer). The connection of WIZ830MJ and base board is supported through 2.54mm Pitch Pin Header typed connector as shown in < Fig 3>

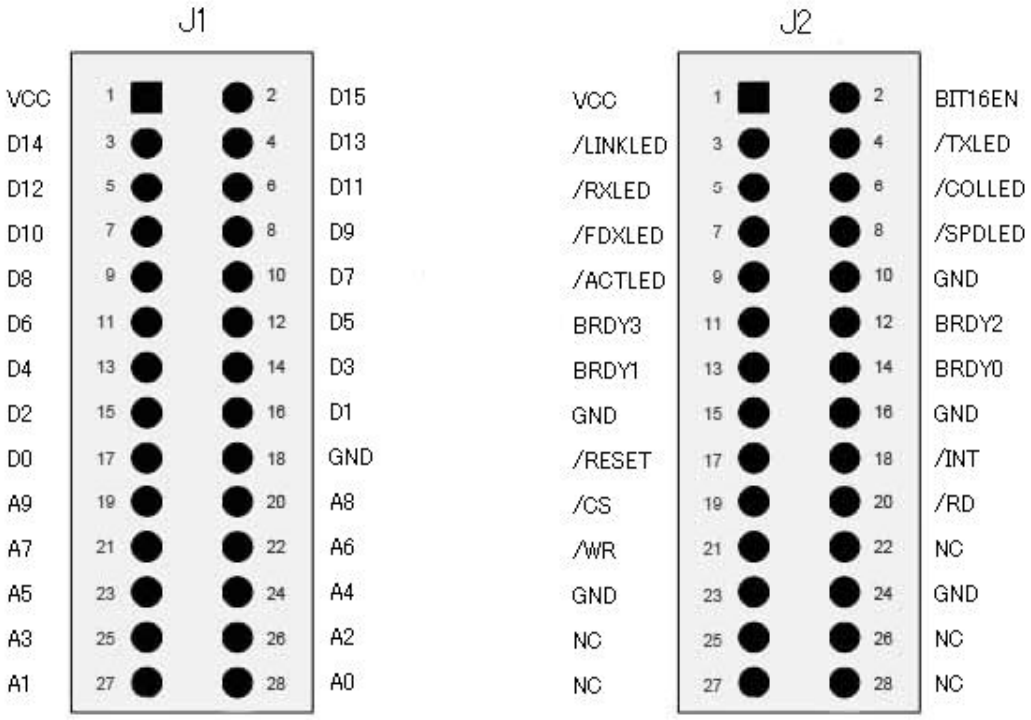


Figure 3-3 : WIZ830MJ Module Interface PIN Map

For more detail, refer to WIZ830MJ Module Datasheet.

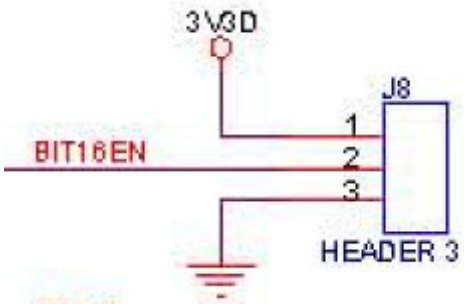


Figure 3-4 : W5300 Data Bus Width Setting Port

By using J8 3Pin header, it is possible to configure data bus width(8bit or 16bit) of W5300 in WIZ830MJ.
By connecting pin 1 and 2 of J8 by using 2 pin jumper, 16 bit data bus width is configured. By connecting pin 2 and 3, 8 bit bus width is configured.

3.2.5. RS-232C Serial Port

It is the interface for UART 0, one of 3 channel UARTs that S3C2410A processor is supporting.

The rest of 2 channels are used for extenstion through expansion port.

W5300E01-ARM platform basically uses 9 pin DSUB male typed connector.

3.2.6. USB Host / Device Port

A-Type Host Connector and Mini-Type Device Connector are provided for testing USB Host interface and USB Device (Slave) interface that S3C2410 supports.

As USB Host driver is basically supported by Linux, it is possible to test various USB devices by connecting to W5300E01-ARM platform. Through USB device driver, the connection with PC is supported.

3.2.7. JTAG Interface

Through JTAG Interface, it is possible to write the Bootloader to the NAND Flash ROM. Debugging is available through JTAG debugging equipment. As 20 pin JTAG connector is installed, general JTAG equipments can be connected without any problem.

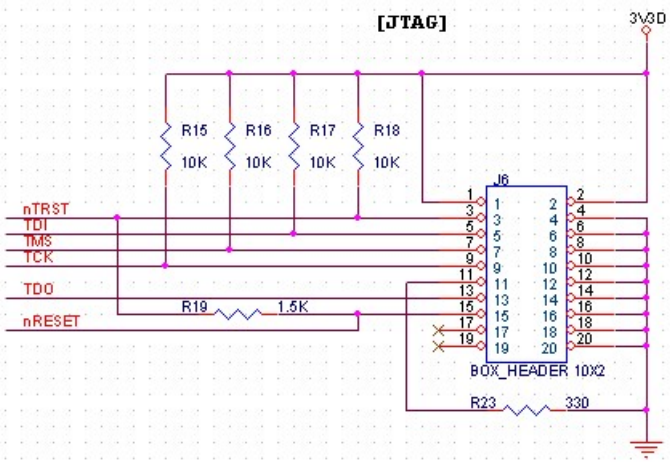


Figure 3-5 : JTAG Interface Part Schematic

3.2.8. LED & Tact Switch for Debugging

By using 2 LEDs and 2 Tact Switches connected to EINT/GPIO, simple debugging is supported.

3.2.9. Character LCD

Character LCD is used for displaying debugging and system staus.

The pin description of character LCD interface (J7) is as below.

PIN#	W5300E01-ARM B/D PIN NAME / LCD PIN NAME	DIR.	Description
1	GND / VSS		Signal Ground
2	5V / VDD	I	LCD Power Supply
3	V0 / V0	I	Voltage for LCD drive
4	A1 / RS	I	Data / Instruction register select
5	A2 / RW	I	Read / Write
6	LCD_E / E	I	Enable signal,start data read / write
7 ~ 14	D0 / DB0 ~ D7 / DB7	I/O	Data Bus Line
15	5V / LED A	O	LED Anode, power supply+
16	GND / LED K	O	LED Cathode,ground 0V

Table 3-1 : LCD PIN Description

74LBC4245 Bidirectional Level shifter is installed between I/O interface voltage level, 3.3V and LCD operational voltage level, 5V, for stable operation.More reliable opration is available by checking LCD Busy Flag through bi-directional buffer.

Low active chip select signal of S3C2410A is passed through inverter and changed to High active.And it is used for LCD Enable singal

For more detail related to LCD operation, refer to LCD datasheet. (**LC1624(R2).pdf).**

3.2.10. Power Block

The power of W5300E01-ARM is supplied by 5V/2A adaptor. The internal power is 5V, 3.3V and 1.8V. For the detail of each power, refer to reference schematic or '3.1.2 Power Block Diagram'.

The input of 5V adaptor can be controlled by power switch (SW1). In order to prevent the damage by over-power when the switch is on, Poly-Fuse(F1) is applied.

Low Drop Out Regulator (5V -> 3.3V, 3.3V -> 1.8V) is applied for power efficiency and heat minimization.

3.2.11. Expansion Port Interface

Expansion port interface is designed for user to add the functions that S3C2410A provides (but W5300E01-ARM does not).

Function	Pin #	Pin Name	Dir.	Description
J3 Port				
Power	1	3V3D		3.3V System power
	3	5V0D		5V System power
	39	GND		System ground
System Data Bus	2,4,6,8,10,12, 14,16,18,20,22, 24,26,28,30,32	D0 ~ D15	IO	Data bus
System Address Bus	5,7,9,11,13,15, 17,19,21,23,25, 27,29,31,33,35	A0 ~ A15	O	Address bus
System Control Signal	34	nGCS0	O	General chip select 0
	36	nOE	O	Output enable
	37	nRESET	I	System reset input
	38	nWE	O	Write enable
	40	EINT0	IO	External interrupt request / GPIO
J4 Port				
Power	1	3V3D		3.3V System power
	3	5V0D		5V System power
	39	GND		System ground
	38	VDDA_ADC		3.3V ADC power
	40	VSSA_ADC		ADC ground
DMA	2	nXDACK0	O	External DMA acknowledge
	4	nXDREQ0	I	External DMA request
I2S Interface	5	I2SLRCK	IO	I2S bus channel select clock
	7	I2SSCLK	IO	I2S bus serial clock
	9	CDCLK	O	CODEC system clock
	11	I2SSDI	I	I2S bus serial data input
	13	I2SSDO	O	I2S bus serial data output
UART Interface	10	TXD1	O	UART1 transmit data output
	12	RXD1	I	UART1 receive data input
	14	TXD2	O	UART2 transmit data output

	16	RXD2	I	UART2 receive data input
SD Card Interface	15	SDCLK	O	SD clock
	17	SDCMD	IO	SD command
	19,21,23,25	SDDAT0 ~ SDDAT3	IO	SD receive / transmit data
SPI Interface	24	SPICLK0	IO	SPI clock
	26	SPIMOSI0	IO	SPI master data output line
	28	SPIMISO0	IO	SPI master data input line
	30	EINT10 / nSS0	I	SPI chip select (for slave mode)
I2C Interface	27	IICSDA	IO	I2C bus data
	29	IICSCL	IO	I2C bus clock
ADC	32	AIN1	AI	ADC analog input 1
	34	AIN0	AI	ADC analog input 0
	36	Vref	AI	ADC voltage reference
GPIO / Interrupt	18	EINT6	IO	External interrupt request / GPIO
	37	EINT1	IO	External interrupt request / GPIO
System Control Signal	6	nXBREQ	I	Bus hold request
	8	nXBACK	O	Bus hold acknowledge
	20	PWREN		
	22	nRSTOUT		
	31	nWAIT		
	33	nGCS4		General chip select 4
	35	nGCS5		General chip select 5
J5 Port				
Power	1	3V3D		
	3	5V0D		
	39	GND		
LCD Data Bus	5,7,9,11,13,15, 17,19,21,23,25, 27,29,31,33,35, 37,2,4,6,8,10, 12,14	VD0 ~ VD23		STN / TFT / SEC TFT LCD data bus
LCD Control Signal	16	LEND		Line end signal
	18	VCLK		LCD clock signal
	20	VLINE		LCD line signal
	22	VM		VM alternates the polarity of the row

				and column voltage
	24	VFRAME		LCD frame signal
	26,28,30	LCDVF0 ~ LCDVF2		Timing control signal for specific TFT LCD (OE/REV/REVB)
GPIO Port / Interrupt	32	EINT12	IO	External Interrupt request / GPIO
	34	EINT23	IO	External Interrupt request / GPIO
	36	EINT22	IO	External Interrupt request / GPIO
	38	EINT21	IO	External Interrupt request / GPIO
	40	EINT20	IO	External Interrupt request / GPIO

Table 3-2 : Expanded Board Interface Pin Description

3.3. Schematic

3.3.1. W5300E01-ARM Base Board Schematic

- Refer to 'W5300E01-ARM_V1.0.DSN' file included in CD.

3.3.2. WIZ830MJ Module Schematic

- Refer to 'WIZ830MJ_R10.DSN' file included in CD.

3.4. Parts List

3.4.1. W5300E01-ARM Parts List

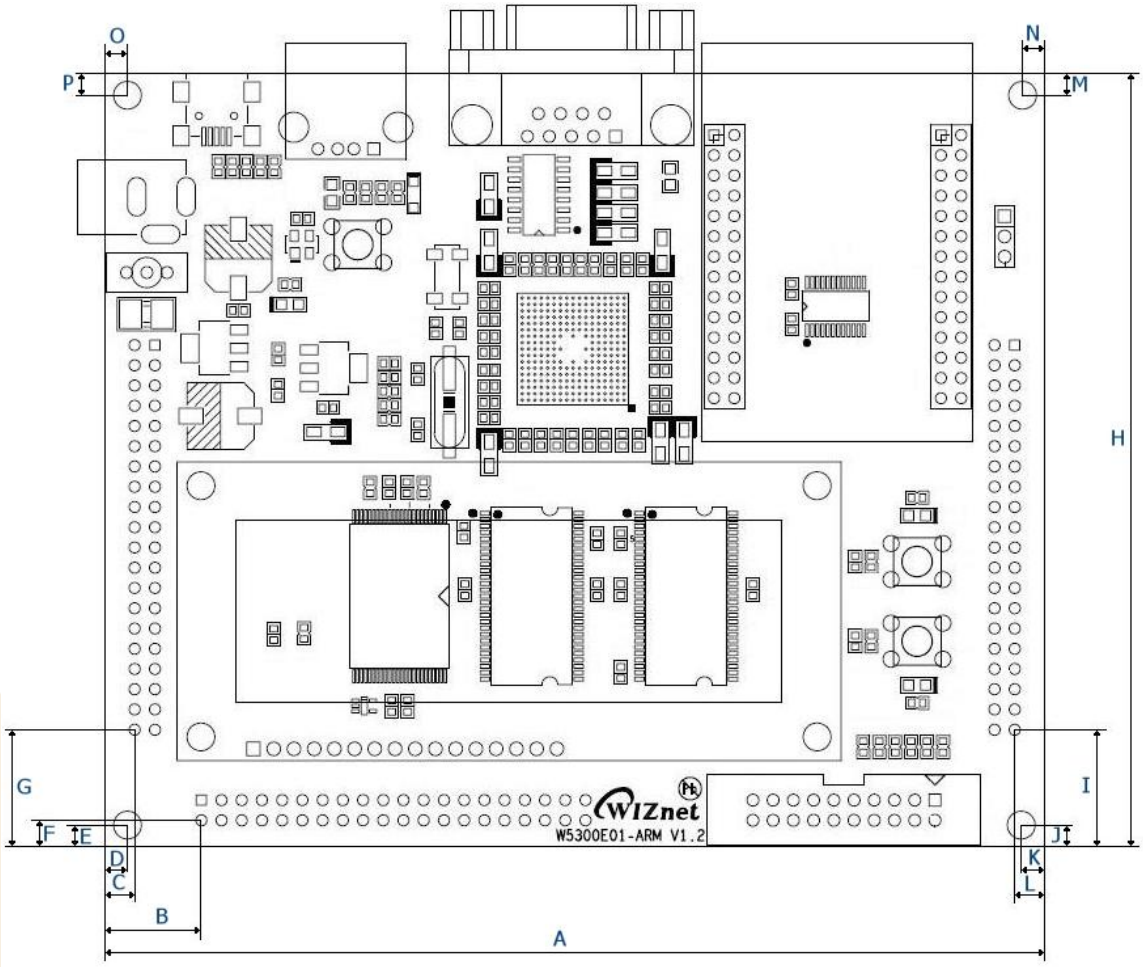
- Refer to '**W5300E01-ARM V1.0 PART LIST.PDF'** file included in CD.

3.4.2. WIZ830MJ Module Parts List

- Refer to '**WIZ830MJ V1.0 PART LIST.PDF'** file included in CD.

3.5. Physical Specification

3.5.1. Board Dimension



Symbols	Dimensions (mm)	Symbols	Dimensions (mm)
A	118.00	I	14.65
B	12.10	J	4.00
C	3.70	K	4.00
D	4.00	L	3.70
E	4.00	M	4.00
F	3.30	N	4.00
G	14.65	O	4.00
H	97.00	P	4.00

Figure 3-6 : W5300E01-ARM Board Dimension

- For the board dimension of WIZ830MJ, refer to WIZ830MJ datasheet.

4. Board Operation

4.1. Booting Check

4.1.1. Booting Check for Windows

Execute the Hyper Terminal of Windows and configure the port as below.

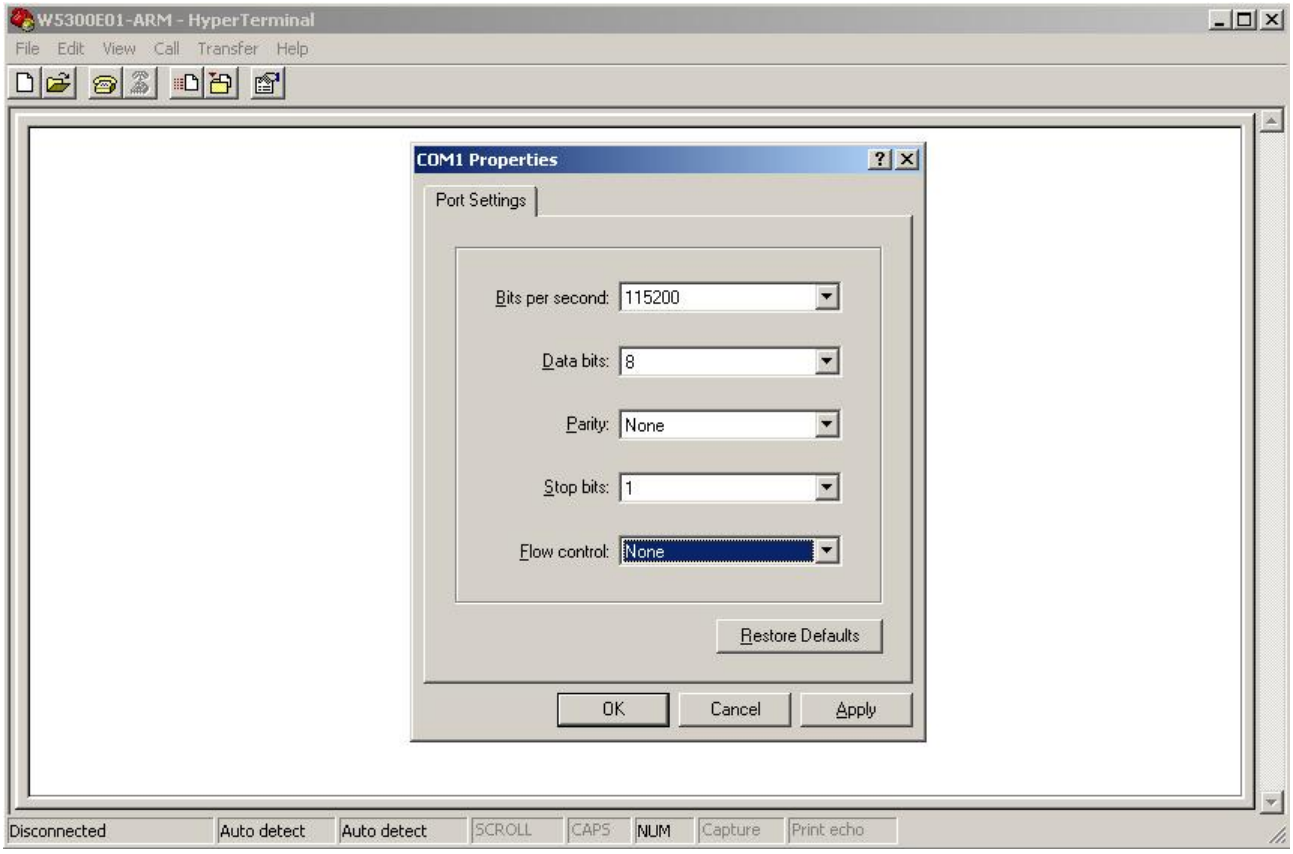


Figure 4-1 : Hyper Terminal Port Configuration

Configure the port as above.

In order to check the operation of the board, turn on the power after connecting the serial cable and executing Hyper Terminal program.

Below booting message means normal operation of the board. Below screen means that the booting is processed to Bootloader and Linux Kernel, and user can use the Linux.

- If below booting message is not displayed, check if power adaptor or serial cable is normally connected.

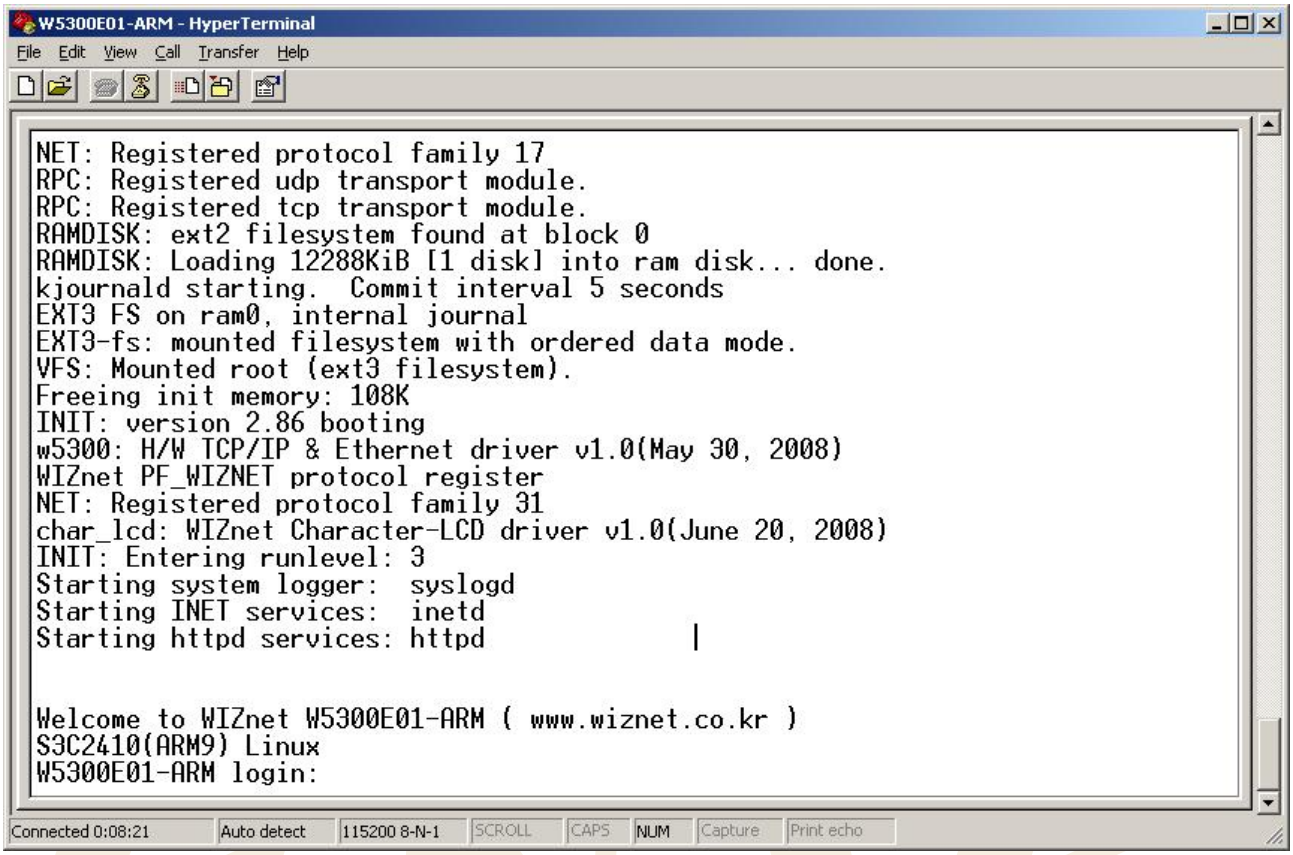


Figure 4-2 : Booting Check at the Hyper Terminal

4.1.2. Booting Check for Linux

At the Linux, the booting can be checked by using minicom.

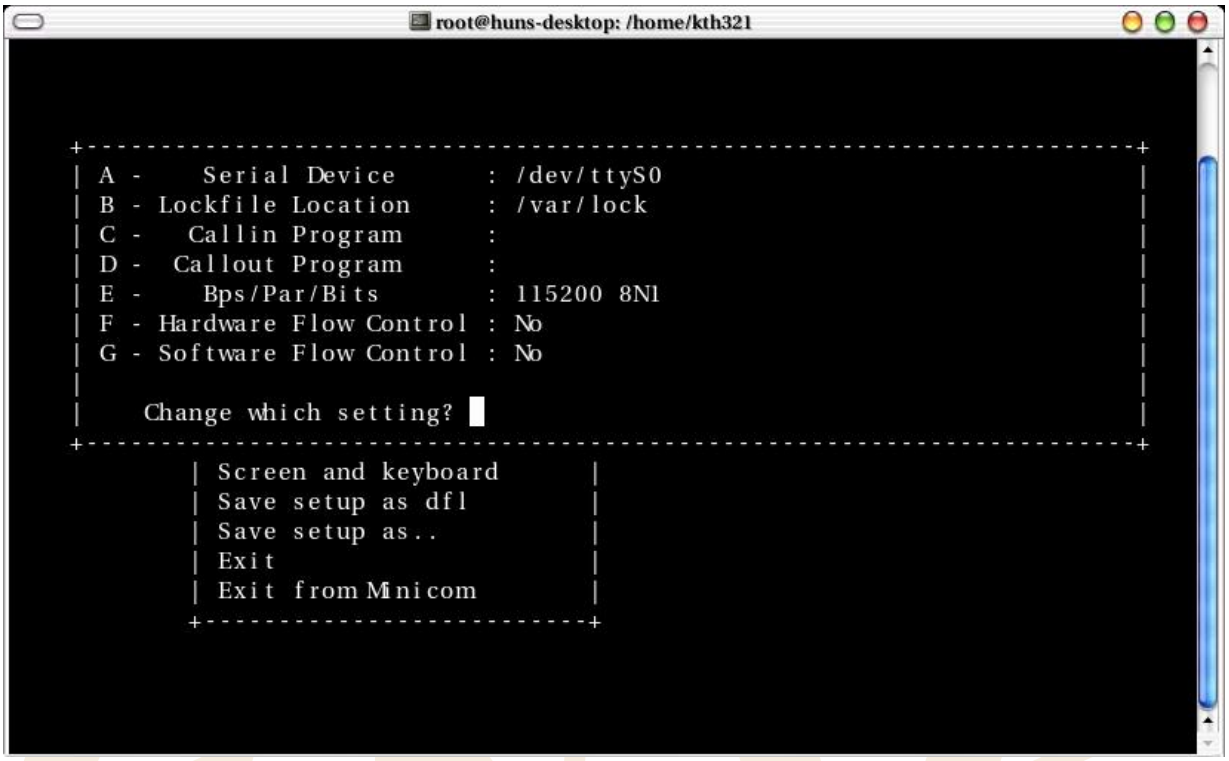


Figure 4-3 : Minicom Port Configuration

Execute 'minicom –s' and select 'Serial port setup' menu, and configure the port as above.
Select 'Save setup as dfl' menu to save serial configuration. By selecting 'Exit' menu, exit the configuration menu.

- The device file of 'Serial Device' can be different according to the Linux version.

Turn on the board after connecting the board to PC by using serial cable, and executing the Hyper Terminal program. If below booting message is displayed on the Terminal screen, the board normally operates.

Below screen means that the booting is processed to Bootloader and Linux Kernel, and user can use the Linux.

- If below booting message is not displayed, check if power adaptor or serial cable is normally connected.

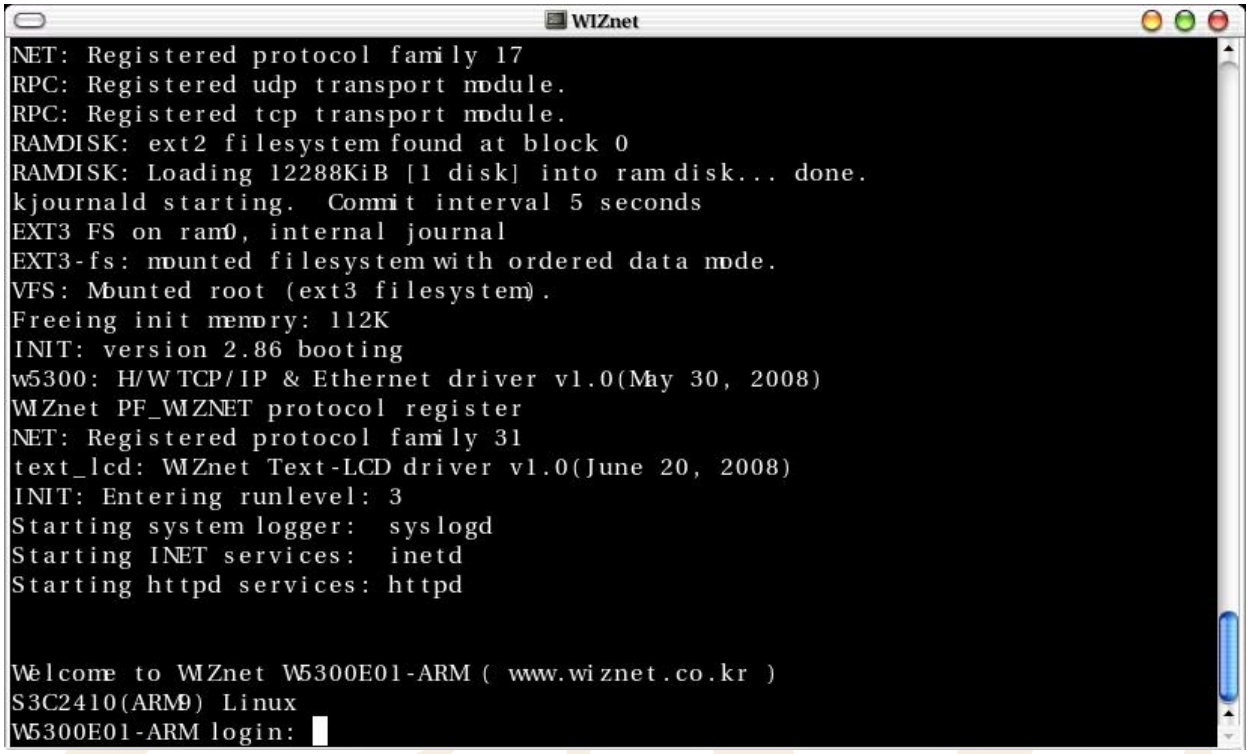


Figure 4-4 : Booting Check at the minicom

4.2. Testing Network Operation

W5300E01-ARM board supports Hybrid mode of W5300 linux driver – simultaneous use of S/W & H/W TCP/IP stack. By configuring channel 0 as MAC_RAW mode, it is possible to utilize Hybrid mode for compatibility with existing network program. Channel 1 is configured for testing looback test through H/W TCP/IP stack.

By configuring channel 0 as MAC_RAW mode, existing network program can be supported without any change. And, if there is any network application requiring higher performance, you can use other H/W TCP/IP channels.

According to the requirement of your products, Hybrid mode can be useful. For more detail, refer to 'W5300 Datasheet'.

When W5300E01-ARM board is booted, FTP server, HTTP server and Telnet server are automatically loaded.

4.2.1. Testing FTP Server

1. Supply the power to W5300E01-ARM and check booting is normally processed.
2. Execute FTP Client program.
   - Windows : Start -> 'Run' -> input 'ftp'
   - Linux : Input 'ftp' at the terminal
3. Input 'open 192.168.1.53' and connect to FTP server of W5300E01-ARM.
4. Log in with 'id > wiznet, password > wiznet'.

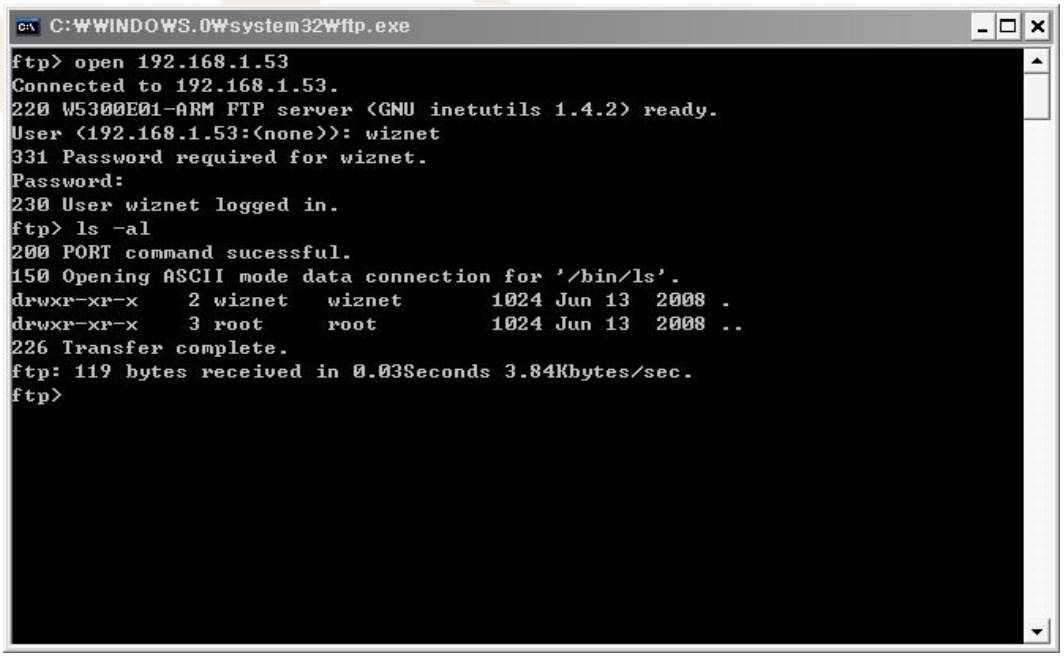


Figure 4-5 : FTP Server Test

4.2.2. Testing HTTP Server

1. Supply the power to W5300E01-ARM board and check booting.
2. Execute Web Browser.
3. Input 'http://192.168.1.53' for address.

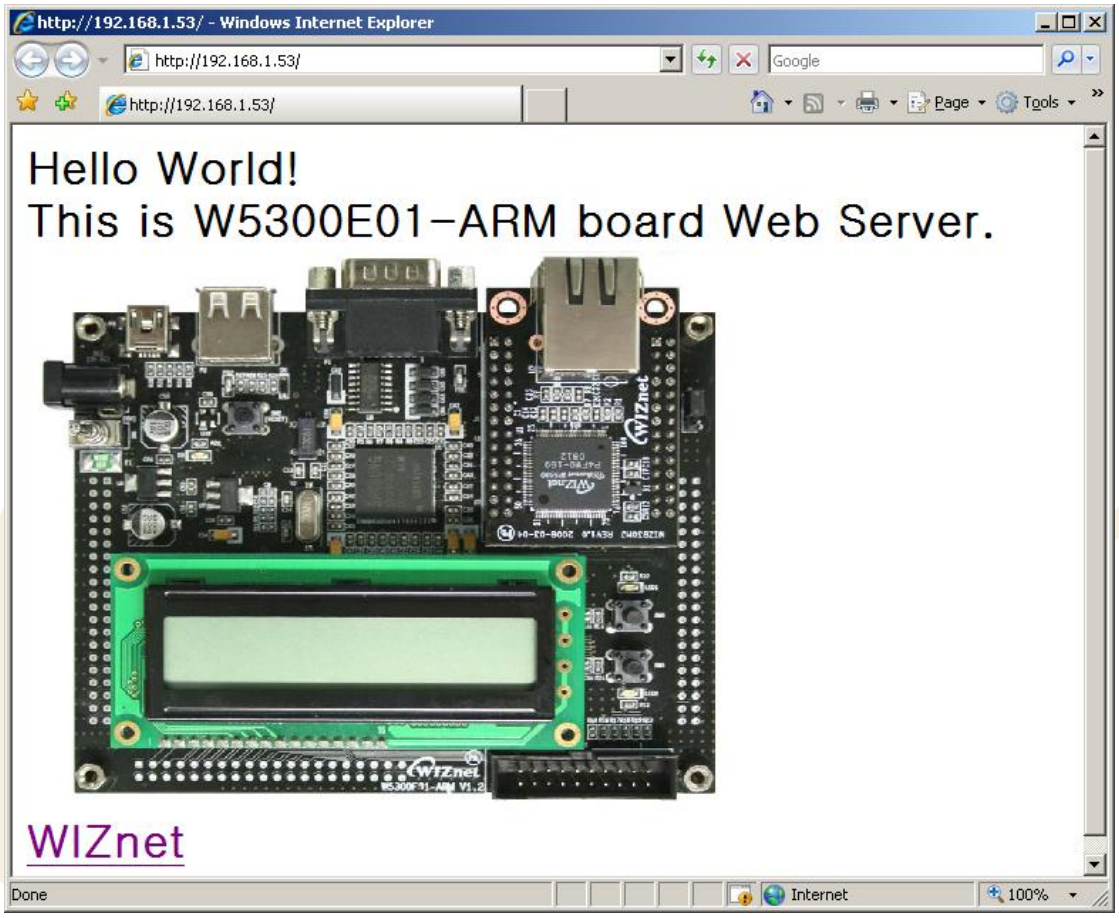


Figure 4-6 : HTTP Server Test

4.2.3. Testing TELNET Server

1. Supply the power to W5300E01-ARM board and check booting.
2. Execute TELNET Client program.
   - Windows : Start -> 'Run' -> Input 'telnet'
   - Linux : Input 'telnet' at the terminal
3. Log in with 'id > wiznet, password > wiznet'.

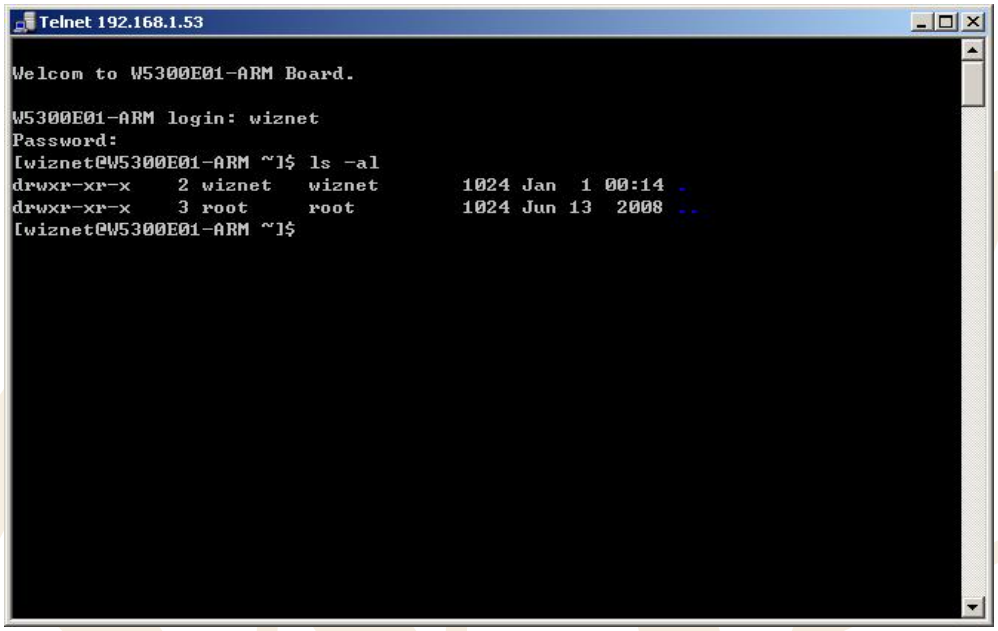


Figure 4-7 : TELNET Server Test

4.2.4. Testing Loopback

With the loopback program in W5300E01-ARM, it is possible to test both of S/W & H/W TCP/IP stack. Loopback program is installed in '/root' directory of 5300E01-ARM board.

1. Install AX1 provided by CD. (**Software/Tools/AXInstallV3.1.exe**)
   For the detail of AX1, refer to '**Documents/Menual/AX1 Menual V3.1'** provided by CD.
2. Supply the power to W5300E01-ARM and check booting.
3. Log in with 'root'.
4. Execute the Loopback server with './loopback –t -w' command. The option of Loopback is as below.

-h	Looback help
-u	UDP loopback mode
-t	TCP loopback mode
-w	PF_WIZNET(H/W TCP/IP) loopback mode(Default 'PF_INET')
-b <size>	Buffer size(Default 4096)
-p <port>	Port number(Default 5001)

Table 4-1 : Loopback Option

5. Execute AX1 at the PC.
6. For the exact testing, calculate CPU Tick by selecting 'CPUTICK' -> 'CPUTICK' of AX1.
7. Select 'TCP' -> 'Connect' of AX1.
8. Input 'IP > 192.168.1.53' and 'PORT > 5001' and click 'OK' button.
   The IP address of PC in which AX1 is installed, should be set as 192.168.1.xxx (Ex> 192.168.1.2).
9. Check if 'Connected' message is shown. If PC and W5300E01-ARM board is connected, 'loopback start!' message is displayed at the loopback program of W5300E01-ARM. If connection is not normal, check below.
   - Is UTP cable correctly connected?
   - Is IP address correctly configured as 192.168.1.xxx?
   - Is loopback program normally executed at the W5300E01-ARM?

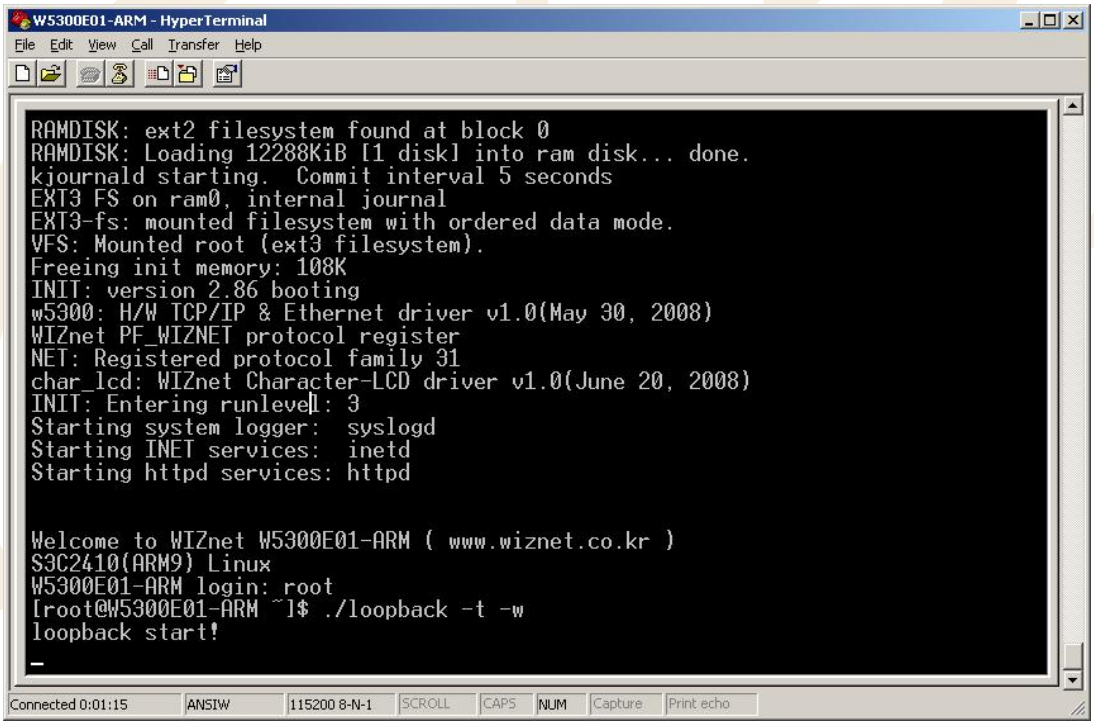


Figure 4-8 : Loopback Connection Check

10. Select '∞' Toolbar of AX1, and select any image file. (Size : 10~50Mb)

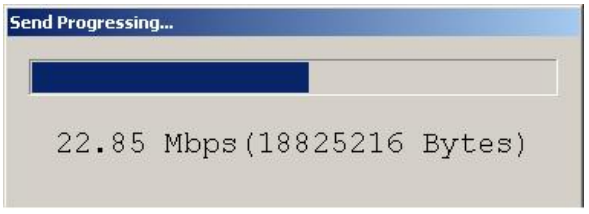


Figure 4-9 : Loopback Test

5. Configuring Development Environment

Development environment is same as one which is generally usede. Programming is done at the Host PC. Execution file for target board is created by using cross compiler. The file is sent to target board for execution.

5.1. Cable Connection

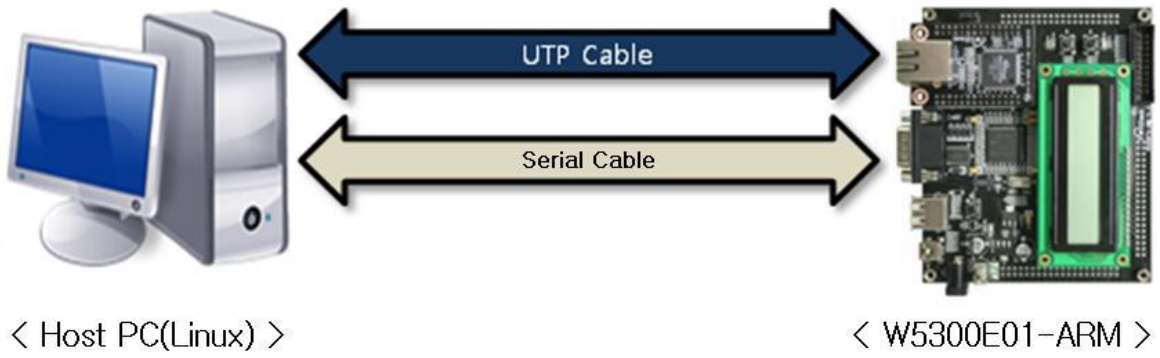


Figure 5-1 : Cable Connection Example 1

As shown in <Figure 5-1>, connect the Host PC and W5300E01-ARM board. The Linux should be installed in the Host PC. Serial cable is used for system console, debugging and data transmission. UTP cable is for data transmission. The data communication through UTP cable is faster than serial cable. In order to transfer big-sized file, use UTP cable for more efficiency.

If you are not familiar with Linux, it is possible to develop at the Windows, and compile the source code by sending to Host PC as shown in <Figure 5-2>.

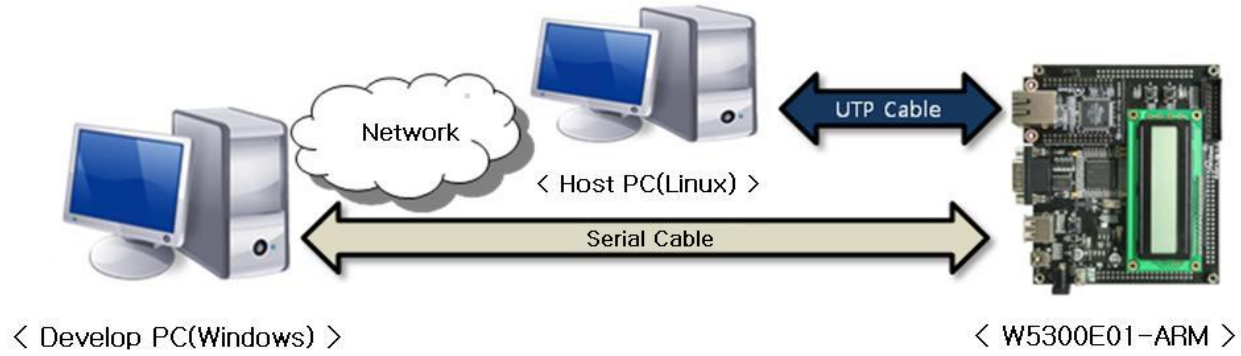


Figure 5-2 : Cable Connection Example 2

5.2. Toolchain Installation

Toolchain is included in CD. Log in with 'root' at the Host PC, and install as below steps.

1. Mount the CD on the CD-ROM.
2. Copy the compressed Toolchain file into '/'.
3. Extract the Toolchain compressed file.

After mounting CD, input below commands in the terminal window.

```
# mount /dev/cdrom /media/cdrom
# cp /media/cdrom/Software/Tools/arm-toolchain-3.4.3.tar.gz /
# cd /
# tar zxvf arm-toolchain-3.4.3.tar.gz
```

As the CD contents are automatically mounted in the latest Linux version, you don't need to input 'mount' command in the most cases. When you mount the CD manually, check device file and mounting directory.

5.3. Network Configuration of Host PC

In order to transfer files between Host PC and target board by using UTP cable, server programs should be installed in the Host PC. Virtual IP address can be used for this networking. As server program installation is processed by package management system, the board should be conneted to the WAN. In order to use LAN and WAN together, add the virtual IP with below command.

```
# ifconfig eth0 add 192.168.1.2
```

5.3.1. TFTP(Trivial File Transfer Protocol) Server Configuration

In order to transmit Linux Kernel image, Ramdisk image or other files, tftp server should be installed in Host PC. In here, we will install the tftp server of Red Hat and Debian.

If Host PC is using Debian Linux, install tftpd-hpa package by using apt-get package management system.

```
# apt-get install tftpd-hpa
```

At the Red Hat Linux, install tftpd-server package by using yum package management system.

```
# yum install tftp-server
```

Make the directory to be used for tftp server, and execute tftp server.

```
# mkdir /tftpboot
# in.tftpd –l –s /tftpboot
```

Now, it is possible to download the files in '/tftpboot' of Host PC into the target board by usig tftp.

When executing tftp server by using xinetd, open '/etc/xinetd.d/tftp' file at the editor, and change the 'disable = yes' into 'disable = no'. If you don't have '/etc/xinetd.d/tftp' file, create the file and include below.

```
service tftp
{
        disable         = no
        socket_type     = dgram
        protocol        = udp
        wait            = yes
        user            = root
        server          = /usr/sbin/in.tftpd
        server_args     = -c –s /tftpboot
        per_source      = 11
        cps             = 100 2
        flags           = IPv4
}
```

Re-start the 'xinetd'

```
# /etc/init.d/xinetd stop
# /etc/init.d/xinetd start
```

When using 'xinetd', 'xinetd' package should be installed. If 'xinetd' package is not installed, install it by using apt-get or yum package management system.

5.3.2. NFS(Network FileSystem) Server Configuration

When transmitting big-sized file from Host to target board, serial is not efficient. For this case, NFS (Network File System) can be used. Binary image is created after programming and compiling at the Host PC. If you install the the binary image at the directory configured by NFS, you can execute it at the target board.

If nfs server is not installed in the Host PC, install the nfs server. If you are using Debian Linux, install the nfs-kernel-server package with apt-get package management system.

```
# apt-get install nfs-kernel-server
```

In case of Red Hat, install nfs-utils package with yum package management system.

```
# yum install nfs-utils
```

Create a directory to share NFS between Host PC and Target board.

```
# mkdir /nfs
```

Open '/etc/'exports' file and add below.

```
/nfs        192.168.1.0(rw, insecure)
```

Re-start the NFS. (In case of using Debian, input '/etc/init.d/nfs-kernel-server' instead of '/etc/init.d/nfs'.)

```
# /etc/init.d/nfs stop
# /etc/init.d/nfs start
```

5.4. File Transmission

5.4.1. File Transmission by using Serial (ZModem)

It is possible to transmit a file by using Zmodem of terminal program.

Serial transmission is simple and easy, but slow in speed. It is appropriate to transmit small-sized files.

For the testing, program 'Hello World' in the 'test.c' source file.

```
#include <stdio.h>

int main(void)
{
        printf("Hello World!\n");
        return 0;
}
```

Compile the source code into execute file for ARM.

```
# arm-linux-gcc –o test test.c
```

5.4.1.1. Serial File Transmission at the Minicom

It is possible to transmit serial data to Minicom by using serial console at the Linux Host PC as below.

1. Exectue minicom, and log in with 'root' after W5300E01-ARM is booted.
2. Press 'Ctrl + A' and 'Ctrl + S', and select ZModem as transmission protocol.

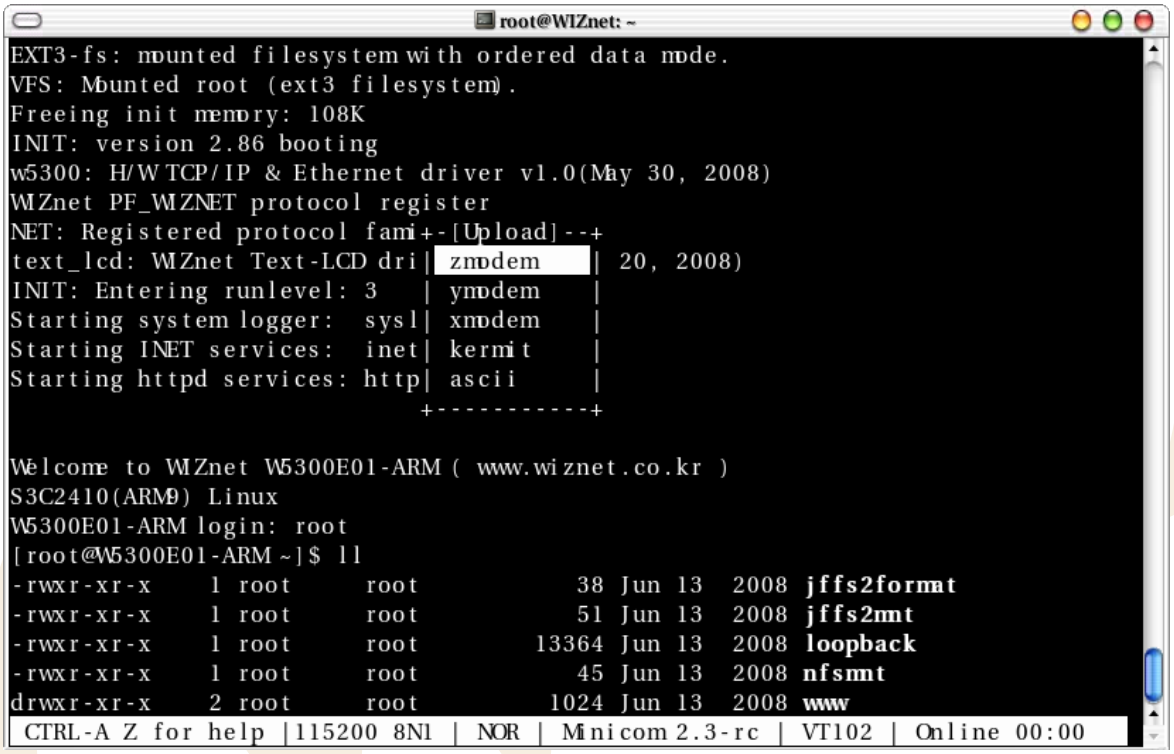


Figure 5-3 : Zmodem Selection at the minicom

3. Move the cursor to the file to be transmitted. If you press the 'Space Bar', the file is highlighted. Now, if you input 'Enter', the file is transmitted. The 'test' file is transmitted.
4. When transmission is completed, below message is displayed.

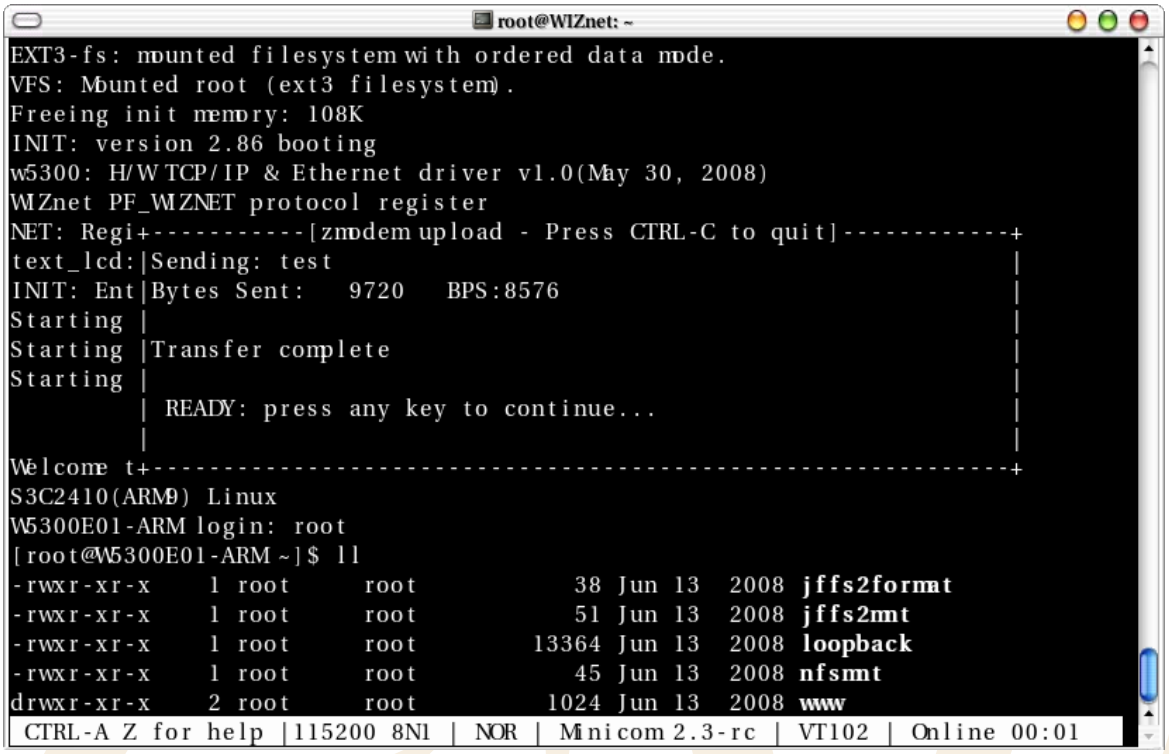


Figure 5-4 : Zmodem at the minicom

5. Check if the file is correctly trnasmitted with 'ls' command.
6. Configure file authority with 'chmod 755 test' and execute 'test' program.

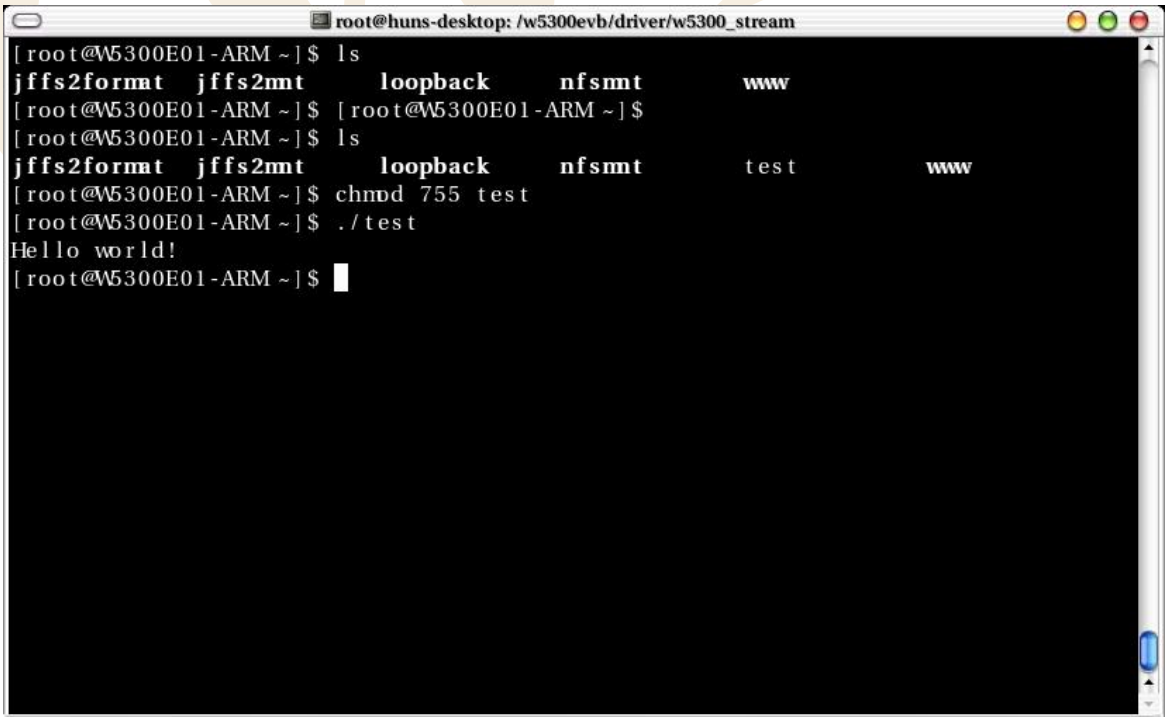


Figure 5-5 : Executing Test File at the minicom

5.4.1.2. Serial File Transmission at the Hyper Terminal

If PC is operating on Windows, the Hyper Terminal can be used as below.

The serial file transmission can be processed at the Hyper Terminal.

1. Execute Hyper Terminal, and log in with 'root' after W5300E01-ARM is booted.
2. Select 'Transfer' -> 'Send File…' at the menu bar.
3. When 'Send File' dialogue window appears, select 'Zmodem with Crash Recovery' or 'Zmodem' for Protocol.

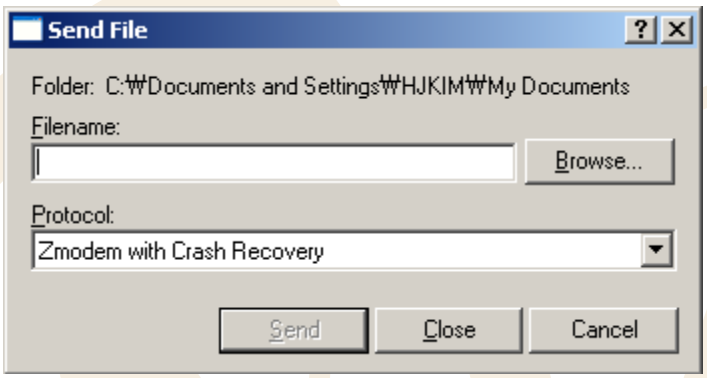


Figure 5-6 : Selecting Zmodem at the Hyper Terminal

4. Click 'Browse…' button for seleting file to be transmitted.
5. If transmission is completed, check if the file is correctly transmitted with 'ls' command.

5.4.2. File Transmission by using NFS

In order to use NFS, NFS Server should be installed in the Host PC. For the detail of NFS server installation, refer to '5.3.2 NFS Server Configuration'.

W5300E01-ARM supports NFS Client. In order to use the directory of NFS at the Host PC, input below command at the target board.

```
# mount –t nfs –o nolock 192.168.1.2:/nfs /mnt/nfs
# ls /mnt/nfs
```

In the '/root' directory, the script to mount NFS is included. By using this script, NFS directory can be simply mounted.

```
#./nfsmnt 192.168.1.2:/nfs
# ls /mnt/nfs
```

5.4.3. Others

In the W5300E01-ARM, FTP server and client are built in. By using FTP, file transmission is available.

6. Linux Kernel Patch & Compile

- The official Linux kernel can't be used in W5300E01-ARM.
- The kernel code should be modified in accordance with W5300E01-ARM platform.
- In the CD, the modified kernel source is included.
- The file to patch Linux official kernel is also included in the CD.

6.1. Linux Kernel Patch

You can use official Linux kernel source (downloadable from http://kernel.org) and patch it.

Download the Linux kernel version 2.6.24.4.

Download the Linux kernel source file and extrat it in the directory '/usr/src/'

```
# mv linux-2.6.24.4.tar.gz /usr/src/
# cd /usr/src
# tar zxvf linux-2.6.24.4.tar.gz
```

Copy the pacth file (provided by CD) and paste it in the directory '/usr/src/' for applying the patch.

```
# cd /media/cdrom/Software/LinuxKernel
# cp patch-w5300e01-v01 /usr/src
# cd /usr/src
# patch –p0 < patch-w5300e01-v01
```

Now, the official Linux kernel is patched to W5300E01-ARM.

If you open the 'patch-w5300e01-v01' by using Editor program, you can check changed parts.

6.2. Linux Kernel Compile

1. Copy Linux kernel source patched to W5300E01-ARM from CD.

```
# cd /media/cdrom/Software/LinuxKernel
# cp linux-2.6.24.4-w5300e01.tar.gz /usr/src/
```

2. Extract the compression and compile it.

```
# cd /usr/src
# tar zxvf linux-2.6.24.4-w5300e01.tar.gz
# cd linux-2.6.24.4
# make wizImage
```

3. Copy the created kernel image into the directory set as tftp server.

```
# cp wizImage /tftpboot/
```

4. Input 'tftp 31000000 wizImage' at the bootloader of the Target board, and download the kernel image. (For the detail of bootloader, refer to '8. Bootloader')

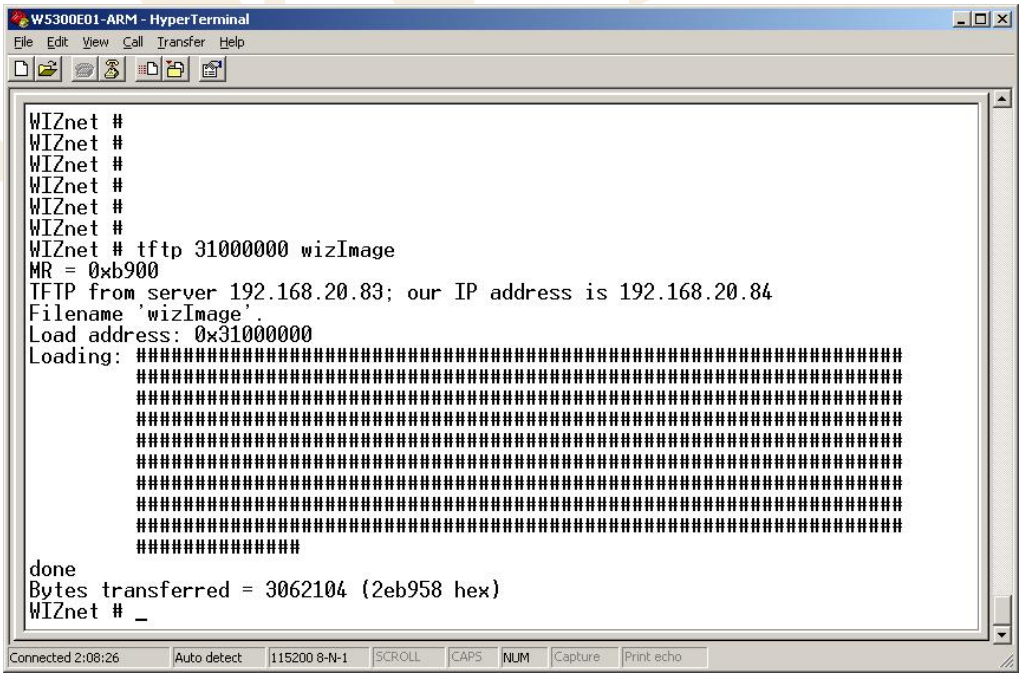


Figure 6-1 : Linux Kernel Image Download

5. Input ‘nand erase 40000 3c0000’ to remove NAND Flash kernel area.
6. Input ‘nand write 31000000 40000 2eb958’ to write the kernel image of RAM into the NAND Flash kernel area. In here, ‘2eb958’ is size of kernel image downloaded from tftp.
7. By inputting ‘printenv’, check bootcmd.
8. When kernel image size is changed, re-configure environment variable ‘bootcmd’
   Ex > If changed kernel image size is **300000**.

WIZnet # setenv bootcmd ‘nand read 30400000 400000 c00000;nand read 32000000 40000 **300000**;bootm 32000000’

9. Input ‘reset’ or check Linux kernel booting by pushing reset button.

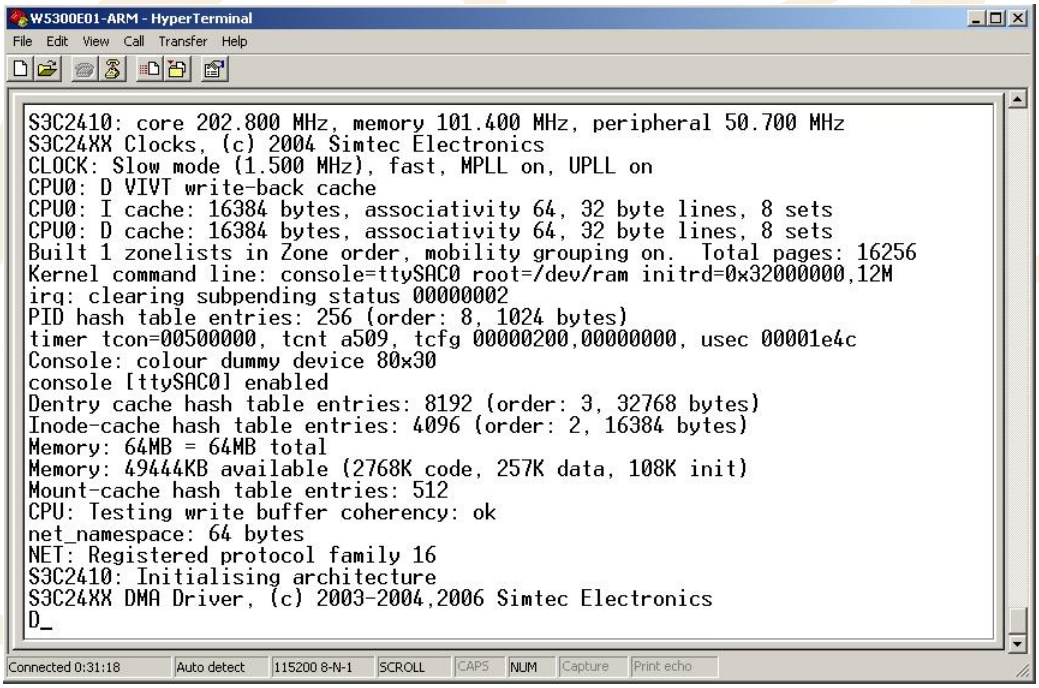


Figure 6-2 : Linux Kernel Booting

7. Root File System

Root File system is the file system mounted in '/'.

Root File system should include dynamic library, device file, and indispensable utilities basically used for system operation.

W5300E01-ARM uses Ramdisk for Root File system. NAND Flash area is used by mounting JFFS2 file system at the Ramdisk

7.1. Ramdisk

Ramdisk uses a part of memory as like disk drive. Ramdisk is volatile because the file system exists on RAM. If power is off, the saved data disappears.

If data should be preserved, the data should be written in NAND Flash area.

(For the detail, refer to '7.2 JFFS2'.)

You can change the Ramdisk composition. However, the size of Ramdisk cant' be over 12M (the maximum size of Ramdisk)

7.1.1. Modifying Ramdisk

1. Copy Ramdisk Image from CD.

```
# cp /media/cdrom/Software/Image/ramdisk_w5300e01 /tftpboot
```

2. Create the directory in which Ramdisk is mounted, and mount it.

```
# mkdir /mnt/ramdisk
# mount –t ext3 –o loop /tftpboot/ramdisk_w5300e01 /mnt/ramdisk
# cd /mnt/ramdisk
# ls
```

3. Modify the file system mounted '/mnt/ramdisk' directory.
4. Unmount it.

```
# cd /
# umount /mnt/ramdisk
```

5. Input 'tftp 31000000 ramdisk_w5300e01' at the bootloader of the target board for downloading Ramdisk image.

(For the detail of bootloader, refer to '8. Bootloader')

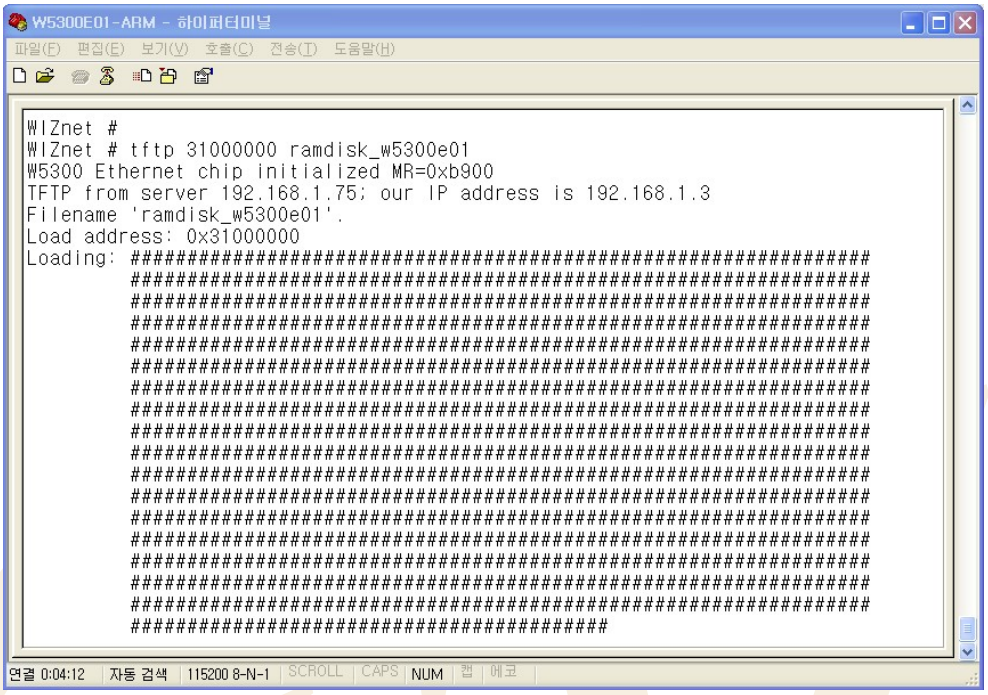


Figure 7-1 : RAMDISK Download

6. Input 'nand erase 400000 1000000' for erasing Ramdisk area of NAND Flash.
7. Input 'nand write 31000000 400000 c00000' for writing Ramdisk image of RAM into Ramdisk area of NAND Flash.
8. Input 'reset' or check if Ramdisk is correctly modified by pushing RESET button.

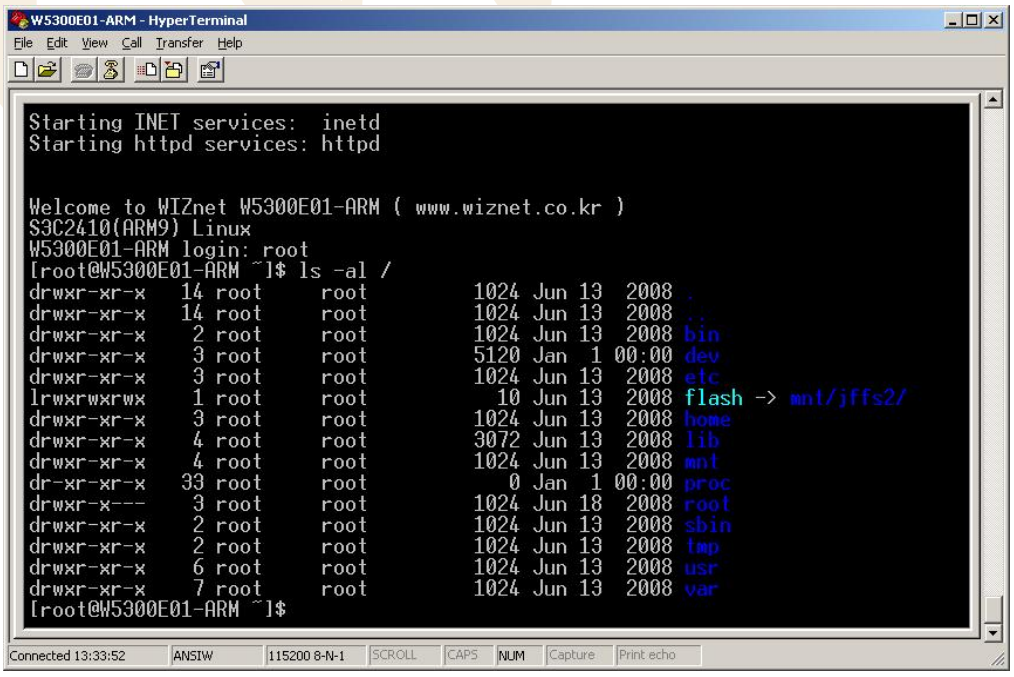


Figure 7-2 : RAMDISK Modification Check

7.1.2. Ramdisk Directory Structure

```
/
|-- bin
|-- dev -- pts
|-- etc -- rc.d
|              |-- init.d
|              |-- rc0.d
|              |-- rc1.d
|              |-- rc2.d
|              |-- rc3.d
|              |-- rc4.d
|              |-- rc5.d
|              `-- rc6.d
|-- flash -> mnt/jffs2/
|-- home -- wiznet
|-- lib
|     |-- ldscripts
|     `-- modules
|-- mnt
|     |-- jffs2
|     `-- nfs
|-- proc
|-- root -- www
|-- sbin
|-- tmp
|-- usr
|     |-- arm-linux – lib -> /lib
|     `-- bin -- include -- sbin
`-- var
       |-- lib
       |-- lock -- subsys
       |-- log
       |-- run
       `-- spool – cron -- crontabs
```

Figure 7-3 : Ramdisk Directory Structure

7.2. JFFS2

NAND Flash is non-volitile storage device. For storing data, use NAND Flash area. JFFS2 is the file system only for Flash. When W5300E01-ARM accesses NAND Flash area, mount and use the JFFS2 file system.
NAND Flahs area can be divided as below

- 0x00000000 – 0x00020000 : Bootloader
- 0x00020000 – 0x00040000 : Boot Param
- 0x00040000 – 0x00400000 : Kernel
- 0x00400000 – 0x01400000 : Ramdisk
- 0x01400000 – 0x04000000 : JFFS2 FileSystem

Bootloader field has bootloader firstly operating when power is supplied to the system.
The Boot Param field is the space where boot parameter value is saved when kernel is booted at the bootloader. Linux kernel field includes kernel images. At the bootloader, the kernel image of this field is loaded to RAM and booted to kernel.
In the Ramdisk area, the Ramdisk used for current root file system is included.
JFFS2 File System area does not have any image. This area can be used after formatting as JFFS2.

7.2.1. JFFS2 File System Mount

1. After completing the boot, log in with 'root'.
2. In case of using JFFS2 area for the first time, it should be formatted in JFFS2 type.

```
# flash_eraseall –j /dev/mtd4
```

3. Mount JFFS2 file system.

```
# mount –t jffs2 /dev/mtdblock4 /mnt/jffs2
```

4. If data is written in the directory - '/mnt/jffs2', the data is saved in NAND Flash.

Now, JFFS2 file system can be automatically mounted.

8. Bootloader

In the W5300E01-ARM, the bootloader that u-boot is modified, is included. The usage of u-boot is almost same as exisiting one.

For the detail of each command, refer to http://www.denx.de/wiki/DULG/Manual. You can also see the information of each command through 'help [command]'

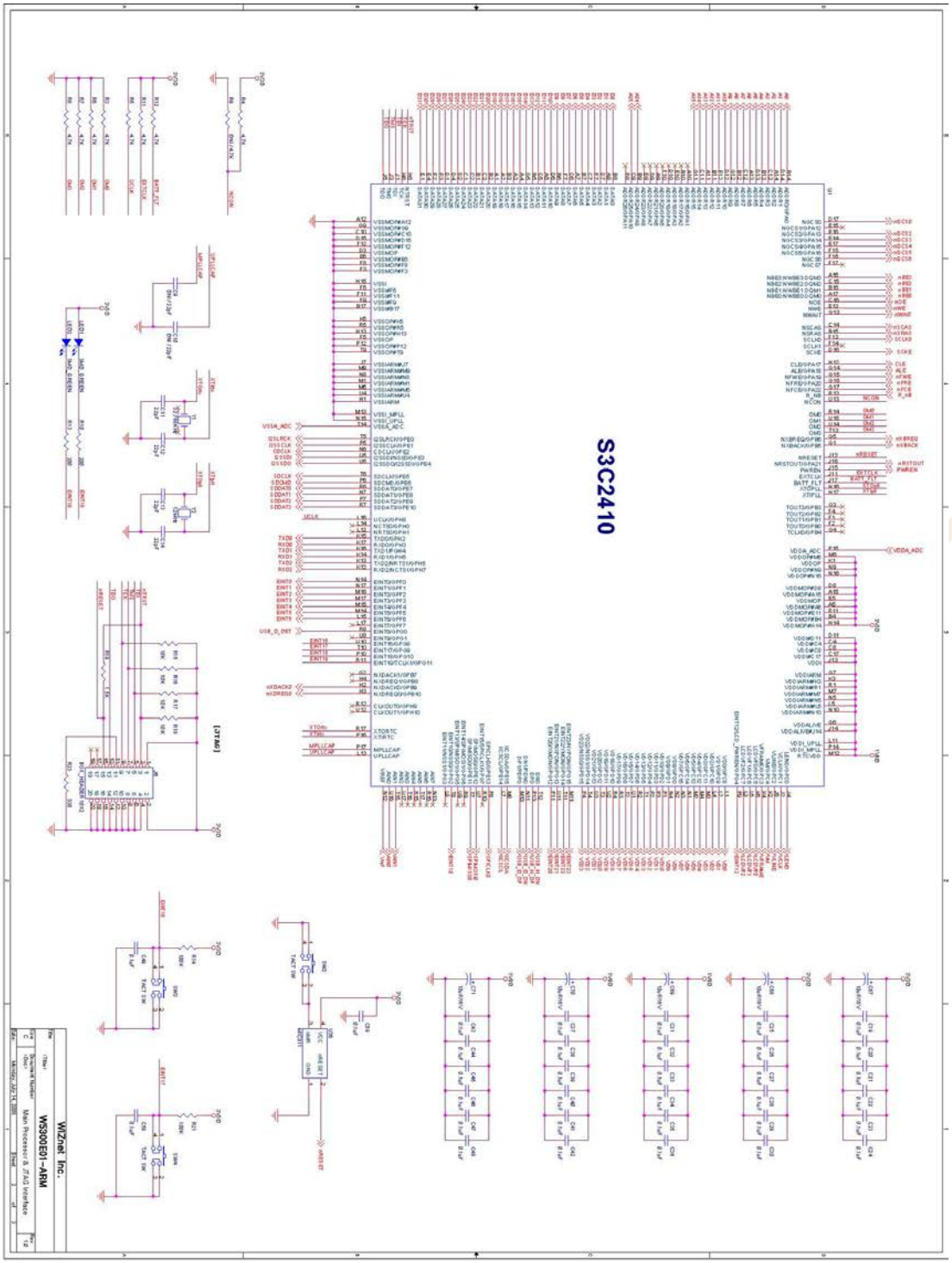
Below is the description of frequently used commands.

printenv	It shows environment variables currently set.
setenv <name> <value>	It configures <name> environment variables as <value>. Some important variables should be configured in accordace with system. ● ipaddr : It shows IP address of the target board during current bootloader operates. Defalut value is '192.168.1.3'. ● serverip : It is the IP address of Host PC. The default value is '192.168.1.2'. ● gatewayip : It is the Gateway IP address. The default value is '192.168.1.1'. ● netmask : It is the Network mask value. The default value is '255.255.255.0'. ● bootcmd : It is the environment variables automatically executed when booting at the bootloader.
tftpboot <address> <bootfilename>	Download the <bootfilename> image file of Host PC of which IP address is set as 'serverip' into <address> of the target board. At this time, tftp server should be installed in the Host PC. For the configuration method of tftp server, refer to '5.3.1. tftp server configuration'. If download is not normally processed, check environment variable 'ipaddr' and 'serverip'.
bootm <address>	Boot with application image saved in <address>.
nand erase [clean] [off size]	Erase Nand Flast as much as [size] at the [off] address. Be careful that all value of NAND Flash can be removed if 'nand erase clean' is used.
nand read <addr> <off> <size>	Read the data of <off> address of NAND Flash as much as <size> to <addr> address.
nand write <addr> <off> <size>	Write the data of <addr> of RAM as much as <size> into <off> of NAND Flash. It is used when writing new bootloader, kernel and Ramdisk into NAND Flash.

Table 8-1 : Bootloader Commands

9. Appendix

9.1. Hardware Schematic of W5300E01-ARM Base Board



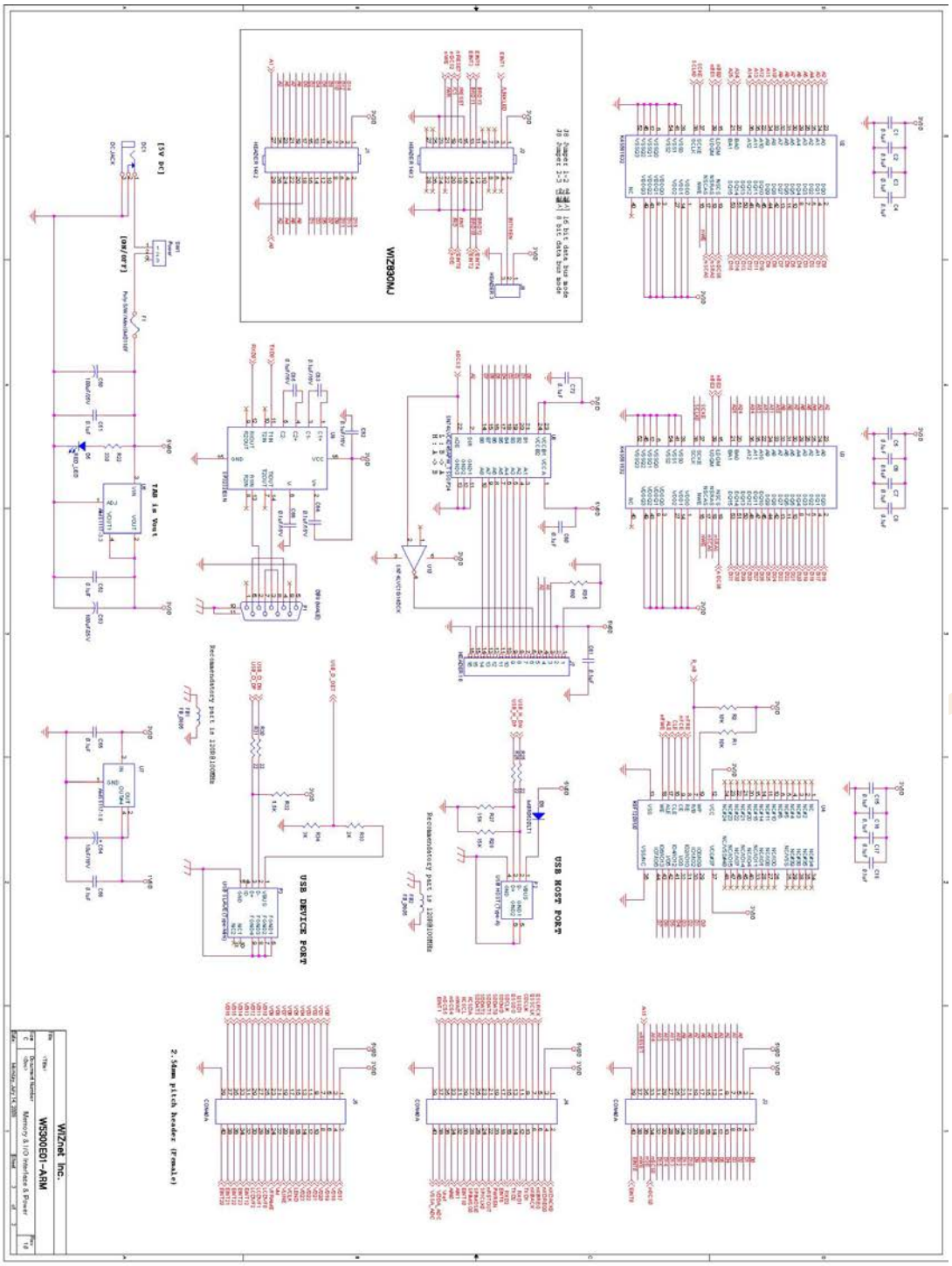


Figure 9-1 : Hardware Schematic of W5300E01-ARM Base Board

- For the schematic of WIZ830MJ Module, refer to **WIZ830MJ_R10.DSN file** included in the CD**.**.

9.2. W5300E01-ARM Base Board Parts List

Item	Q.ty	Reference	Part	Tech. Characteristics	Package
1	52	C1,C2,C3,C4,C5 C6,C7,C8,C15,C16,C17,C18,C19,C20, C21,C22,C23,C24, C25,C26,C27,C28, C29,C30,C31,C32, C33,C34,C35,C36, C37,C38,C39,C40, C41,C42,C43,C44, C45,C46,C47,C48, C49,C51,C52,C55, C56,C58,C59,C60, C61,C72	0.1uF	50V-20% Ceramin	CASE 0603
2	4	C11,C12,C13,C14	22pF	50V-20% Ceramin	CASE 0603
3	2	C50,C53	100uF/25V	Aluminum 25Vmin 20%	
4	6	C54,C67,C68,C69, C70,C71	10uF/16V	Tantal 16Vmin 10%	EIA/IECQ 3216
5	5	C62,C63,C64,C65, C66	0.1uF/16V	Tantal 16Vmin 10%	EIA/IECQ 3216
6	1	DC1	DC-Jack	Ø3.5	DIP
7	1	D6	MBR0520LT1	SBR DIODE	SOD-123
8	2	FB1,FB2	120 ohm Chip Ferrite Bead	120 ohm @ 100MHz	CASE 0805
9	1	F1	MiniSMD110F/16	Poly-Fuse	SMD4532
10	2	J1,J2	14x2 Pin Header (F)	2.54mm pitch	2.54pitch, DIP
11	1	J6	10x2 Box Socket	2.54mm pitch	2.54pitch, DIP
12	1	J7	16x1 Pin Header (F)	2.54mm pitch	2.54pitch, DIP
13	1	J8	3x1 Pin Header (M)	2.54mm pitch	2.54pitch, DIP
14	2	LED1,LED2	Chip LED, Green		CASE 0805
15	1	D5	Chip LED, RED		CASE 0805
16	1	P1	DSUB 9Pin Male, R/A	RS-232 Serial Connector	DIP
17	1	P2	USB A-Type Connector	USB Host Connector	DIP

18	1	P3	USB Mini-Type Connector	USB Device(Slave) Connector	SMD
Item	**Q.ty**	**Reference**	**Part**	**Tech. Characteristics**	**Package**
19	6	R1,R2,R15,R16, R17,R18	10K ohm	1/10W-5% SMD	CASE 0603
20	8	R3,R4,R5,R6,R7, R8R11,R12	4.7K ohm	1/10W-5% SMD	CASE 0603
21	2	R10,R13	200 ohm	1/10W-5% SMD	CASE 0603
22	2	R22,R23	330 ohm	1/10W-5% SMD	CASE 0603
23	2	R14,R21	100K ohm	1/10W-5% SMD	CASE 0603
24	2	R19,R32	1.5K ohm	1/10W-5% SMD	CASE 0603
25	4	R25,R26,R30,R31	22 ohm	1/10W-5% SMD	CASE 0603
26	2	R27,R28	15K ohm	1/10W-5% SMD	CASE 0603
27	1	R33	2K ohm	1/10W-5% SMD	CASE 0603
28	1	R34	3K ohm	1/10W-5% SMD	CASE 0603
29	1	R35	680 ohm	1/10W-5% SMD	CASE 0603
30	1	SW1	3Pin Power Toggle S/W	Power Switch	DIP
31	3	SW2,SW3,SW4	STS-110B	4Pin Tact Switch	DIP
32	1	U1	S3C2410A	SAMSUNG ARM920T Processor	272-FBGA
33	2	U2,U3	K4S561632H	SDRAM 32MB	TSOP(II)
34	1	U4	K9F1208U	NAND Flash ROM 64MB	TSOP(I)
35	1	U6	AMS1117-3.3	LDO 3.3V Output	SOT-223
36	1	U7	AMS1117-1.8	LDO 1.8V Output	SOT-223
37	1	U8	SN74LVC4245APW	Bidirectional Buffer(Level Shifter)	TSSOP24
38	1	U9	SP3232ESN	2-CH RS-232 Transceiver	NSOIC
39	1	U10	SN74LVC1G14DCK	Schumitt Trigger Inverter	SC-70
40	1	U36	MIC811RU	Reset IC (Active Low Output)	SOT-143
41	1	Y1	32.768KHz	Crystal	SMD
42	1	Y2	12MHz	Crystal	SMD
43	1	Character LCD	LC1624	16character * 2line	
44	1	PCB	W5300E01-ARM Rev1.2 / 1.6T, 6-Layer		

Table 9-1 : W5300E01-ARM Base Board Part List

- For the part list of WIZ830MJ Module, refer to '**WIZ830MJ V1.0 PART LIST.PDF**' file included in CD**.**