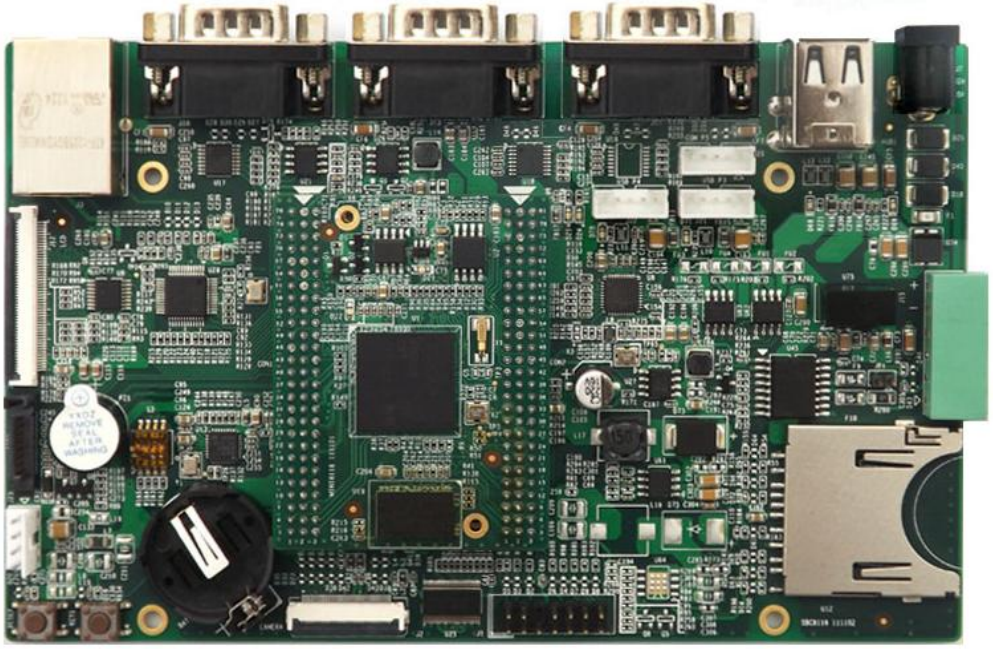
SBC8118

Integrated with SATA、SD、USB、RS485、Ethernet、LCD、CCD/COMS、PRU SUART、JTAG interface based on 32-bit microcontroller AM1808 industrial-grade single board computer



Quick User Manual

Copyright

- SBC8118, CAM8000-A, CAM8000-D, GPRS8000-S, GPS8000-S, WCDMA8000-U, CDMA8000-U, WF8000-U, CAM8100-U, VGA8000 are authorized by Embest Technology Co., Ltd.
- AM1808 are authorized by TI.
- Sourcery G++ Lite for ARM GNU/Linux is authorized by Codesourcery.
- Microsoft, MS-DOS, Windows, Windows95, Windows98, Windows2000 and Windows embedded CE 6.0 are authorized by Microsoft.

Important statements

Version update records:

Version	Date	Description
1.0	2012.9.1	Original version

Contact:

If you want to order products from Embest, please contact Marketing Department:

Tel: +86-755-25635626-863/865/866/867/868

Fax: +86-755-25635626-666

E-mail: market@embedinfo.com

If you want to get technical assistance from Embest, please contact Technical Assistance Department:

Tel: +86-755-25635626-872/875/897

E-mail: support@embedinfo.com

URL: http://www.armkits.com or http://www.embest-tech.com

Address: Tower B 4/F, Shanshui Building, Nanshan Yungu Innovation Industry Park, Liuxian Ave. No. 1183, Taoyuan St., Nanshan District, Shenzhen, China (518055)

Contents

Chapter 1 Overview

The primary purpose of this Document is to quickly understand the hardware and software development environment for SBC8118 single board computer, and help the user to get into the product development faster.

1) This Document mainly has the following sections:

a) How to use documents and CD data quickly

b) Single board computer hardware components and suite configuration

c) Single board computer hardware and software default configurations

d) Quick setup of the development environment

e) Hands-on and quick use of Linux/wince operating systems

2) Factory default configuration of SBC8118 single board computer

a) Hardware Configuration: no RTC cell (Cell: ML2032)

b) Software factory default parameters

- SBC8118 single board computer preferably boots from serial port, NAND Flash and SPI Flash, and support update image via TF card, net port
- NAND Flash is installed by default with linux2.6.33 operating system and 4.3-inch screen driver

3) List of option add-on modules for SBC8118 single board computer:

Modules	Linux	WinCE	Notes
WF8000-U	NO	YES#	Provided with CD-ROM Separately
CAM8000-A	YES*	YES*	Provided with CD-ROM on Development board
CDMA8000-U	NO	YES#	Download here
WCDMA8000-U	NO	YES#	Download here

Table 1-1

* = Provided with Source Code # = Not Provided with Source Code

1.1 Getting Started Quickly

In this section user will learn how to understand and use SBC8118 single board computer efficiently and faster using SBC8118 single board computer Quick User Manual. For more information please refer to the document and location listed below:

For hardware development:

Hardware system	Introduce CPU, expanded chip and hardware interface	User Manual->Chapter 2 Hardware System
CPU Datasheet	Know principle and configuration	CD->\HW design\datasheet\CPU\
Schematic diagram	Know hardware principle	CD->\HW design\schematic
Dimensional drawing	Refer to the actual dimension to bring convenience for opening die	User Manual->Appendix-> Appendix I Hardware Dimensions

Table 1-2

For software development:

Establish testing environment	To connect with external hardware devices, set serial port terminals and boots the system	Quick Operation Manual->2 Quick establishment of development environment 3 Handover and quick use of operating system
Test functionality of interface	Test the interface of the board carrier through the operating system	User Manual ->3.8.2 Various Tests scenario
Establish developing and compilation environment	Linux developing and compilation environment	User Manual -> 3.4.1 Establishing operating system development environment
	WinCE developing and compilation environment	User Manual->4.4.1 Installation of IDE 4.4.2 Extract BSP and project files to IDE
Recompile system image	Recompile Linux system image	User Manual->3.4.2 System Compilation
	Recompile WinCE system image	User Manual->4.4.3 Sysgen & Build BSP
Software development	Refer to introduction of Linux application development process	User Manual->3.7 The Development Of Application

	Refer to introduction of WinCE driver and related driver development process	User Manual -> 4.4.4 Source code path of all drivers in BSP:

Table 1-3

For Marketing:

Hardware system	CPU feature, board carrier interface data	User Manual->Chapter 2 Hardware System
About Linux / WinCE software	Know basic Linux software components and features, and purpose of compilation tool	User Manual->3.2 Software Resources 3.3 Software Features
	Know basic WinCE software components and features, and purpose of compilation tool	User Manual->4.2 Software Resources 4.3 Software Features
Dimensional drawing	Refer to the actual dimension to bring convenience for opening die	User Manual->Appendix-> Appendix I Hardware Dimensions

Table 1-4

For learning personnel:

It is suggested to browse all the sections in each chapter of this Manual..

1.2 Development Kit Content

SBC8118 single board computer is available in two bundles; Standard and Full bundle, the kit contents for each bundle is given below:

Standard Bundle

- SBC8118 single board computer
- Serial port line (DB9-DB9)
- 5V@2A power adapter

Full Bundle

- Standard bundle of SBC8118 single board computer
- 4.3-inch LCD display screen or 7-inch display screen (with touch screen)

DVD/CD Contains:

- SBC8118 Quick Operation Manual
- SBC8118 User Manual
- Schematic Diagram, Board Carrier Chip Datasheet
- Development Kit Software (Linux/WinCE)

Chapter 2 Quick Setup of Development Environment

2.1 Hardware Setup

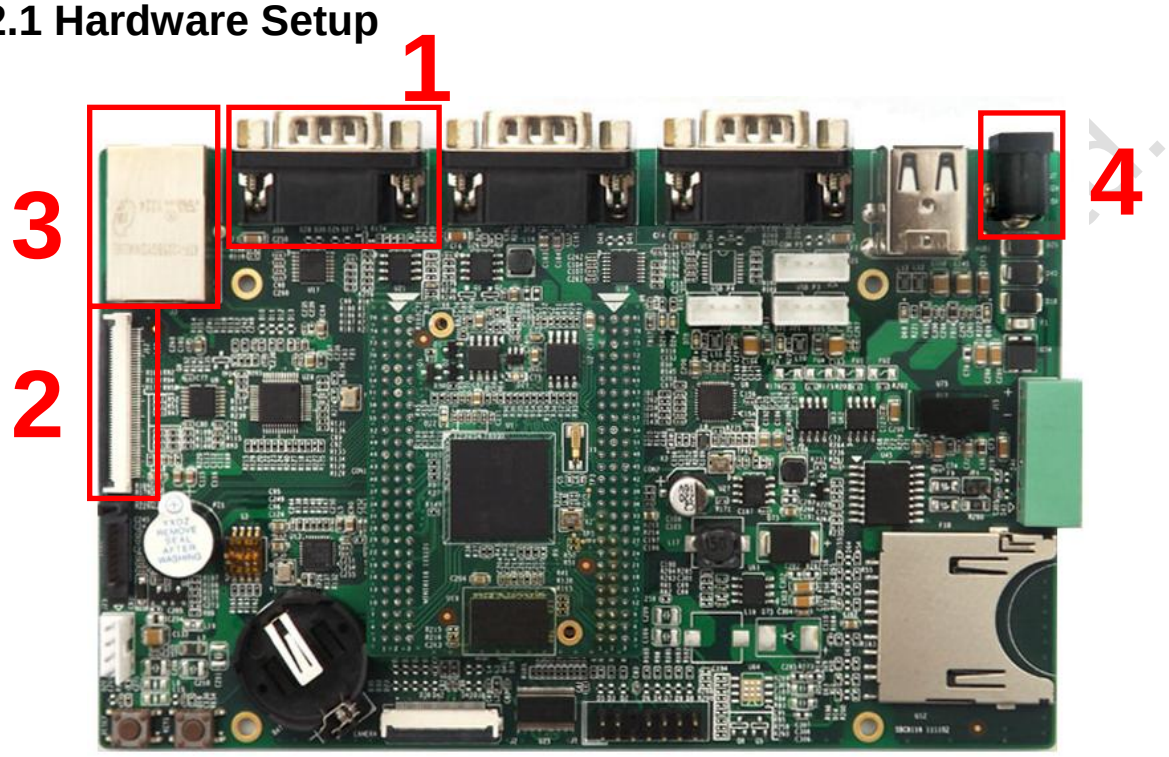


Figure 2-1

Please follow the below steps for hardware setup:

1) Connect serial port for communication

Use serial cable to connect the debugger serial port, PC serial port and SBC8118 serial port.

2) Connect TFT-LCD

Connect your 4.3-inch/7-inch TFT-LCD to the TFT-LCD interface.

Notice: When connect 50-pin FPC cable of LCD to the board, please pay attention to the blue side up, do not plug partial, and not hot-pluggable, so as not to damage the board.

3) Connect Ethernet cable

Connect the Ethernet cable to the position 3 as shown in above Figure 2-1.

4) Connect the 12V power adapter to the single board computer

2.2 Software Setup

2.2.1 Windows XP Setup

1) Setup Linux boot tool

Install Linux boot tool [AISgen_d800k006_Install_v1.7.exe], location on CD: CDROM\Linux\tools\, select the default configuration to install.

2) Setup WinCE component

Install WinCE boot tool [dotNetFx40_Full_x86_x64.exe], location on CD: CDROM\WINCE600\tools\, select the default configuration to install.

3) Setup a HyperTerminal on PC

Before SBC8118 board boot-up, you need to setup a HyperTerminal connection on PC; follow the below steps in order to setup a Hyper Terminal connection:

a) Windows XP -> Start -> All Programs -> Accessories -> Communication -> Hyper Terminal:

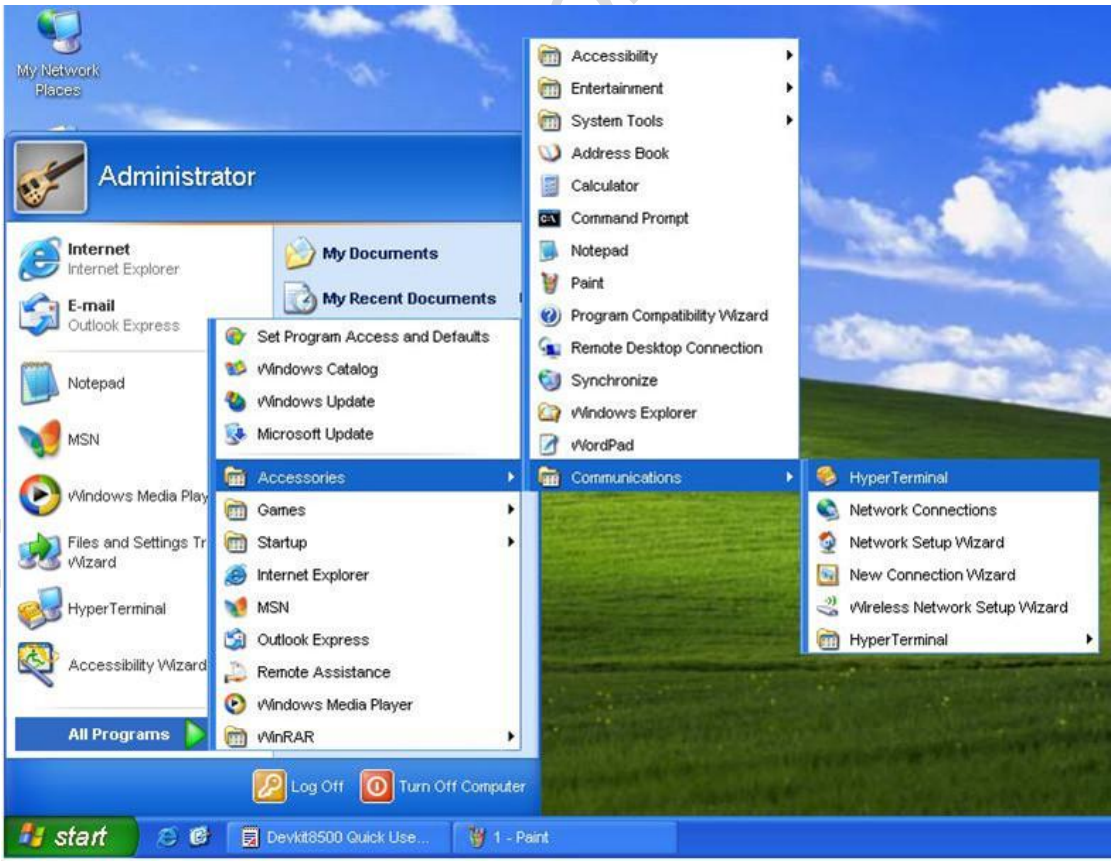


Figure 2-2

b) Establish HyperTerminal connection and give commands:

Figure 2-3

c) Select the specific serial port from the list as per your computer COM port configuration:

Figure 2-4

d) Set parameters for serial port connection as below:

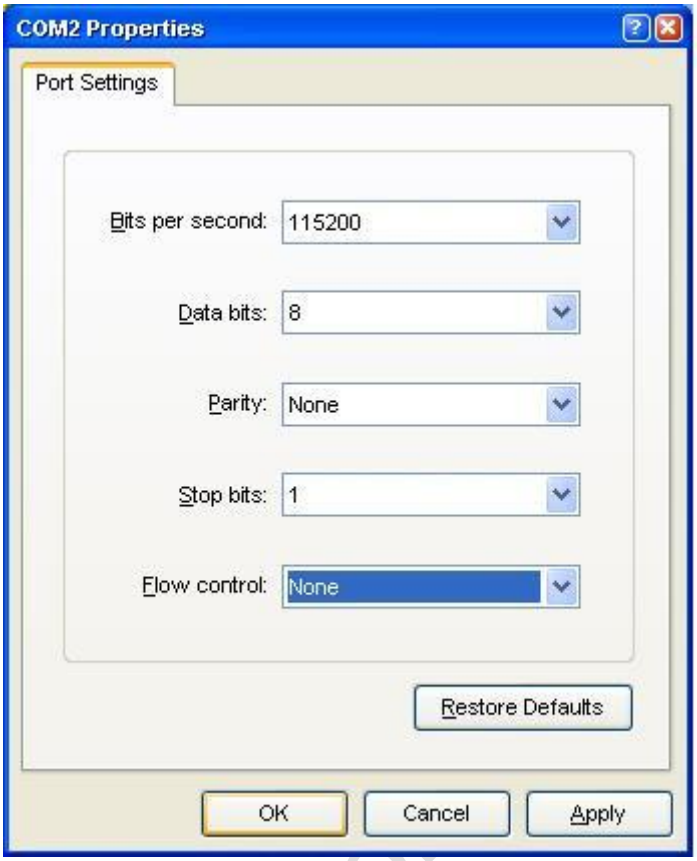


Figure 2-5

e) So we have successfully established a Hyper Terminal connection with PC serial port:

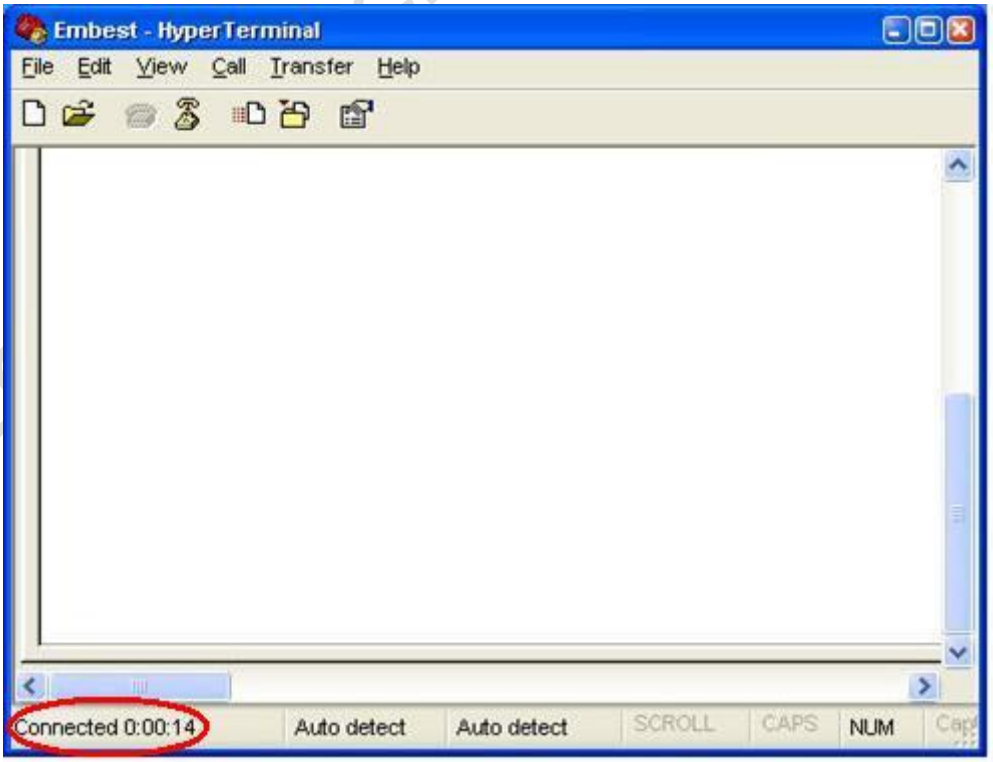


Figure 2-6

Chapter 3 Hands-on and Quick use of Operating System

SBC8118 single board computer supports two operating systems, Linux 2.6.33 and WinCE6.0. This chapter will introduces the method of switching between Linux and WinCE operating system.

3.1 Quick Start up with Linux system

SBC8118 single board computer by default comes with Linux + 4.3-inch screen display installed in NAND Flash. It will boot directly once it's powered on or reset, and to enter into the Linux system you just need to enter "**root**".

Please make sure the switch [S3] settings are selected as below:

Switch S3-1 to ON position, the other bits code of S3 switch to OFF position.

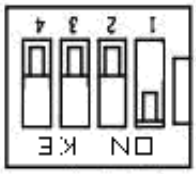


Figure 3-1 Boot-up from NAND Flash

At the factory, the UBOOT is programmed into the SPI Flash too, if you want to boot from SPI Flash, simply set the switch S3 state as follows:

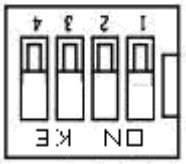


Figure 3-2 Boot-up from SPI Flash

Notice: If you purchased a 7-inch screen, you need to modify the UBOOT parameters, specific methods please refer to the SBC8118 user manual section 3.8.1

3.1.1 Boot-up from Serial port

1) After the setup of hardware, make sure the switch [S3] settings are selected as below:

   Switch S3-3 and S3-4 to ON position, rest switches are on OFF position.

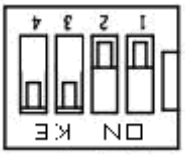


Figure 3-3 Boot-up from Serial port

2) Open the AISgen_d800k006_Install_v1.7.exe:

- Windows XP -> Start -> All Programs -> Texas Instruments -> AISgen for D800K006 -> UART Boot Host
- Add u-boot-uart-ais.bin [Directory: CD\linux\image\] to the “AIS-File”, and change COM port if required, as shown in the following figure:

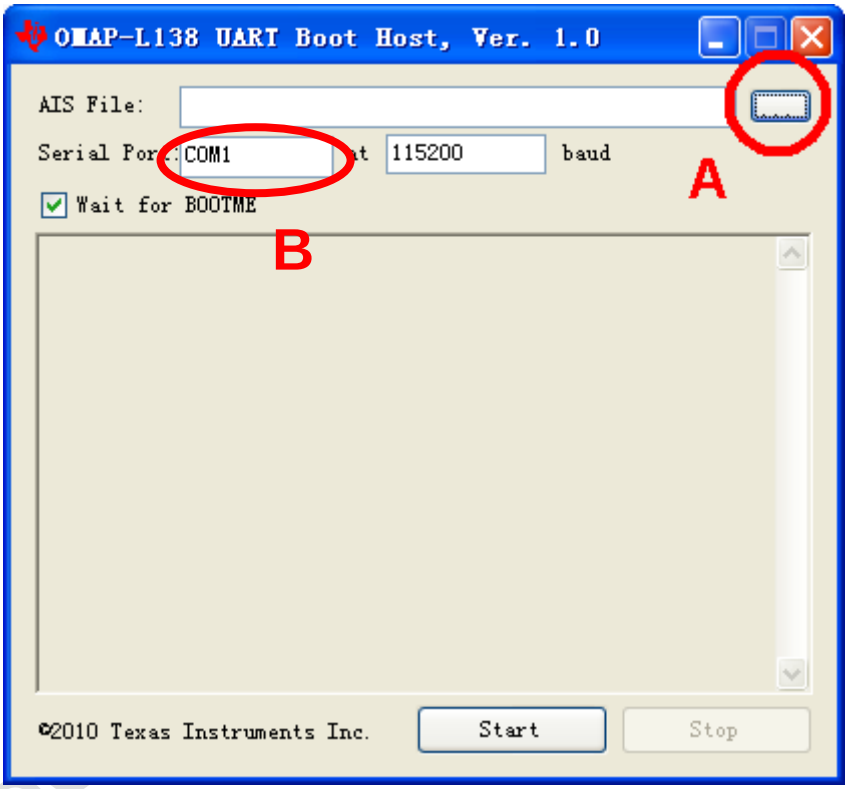


Figure 3-4

3) Click the “Start” and power on the single board computer to boot-up from serial port.

4) Wait for moment, the target window will display “(Serial Port): Closing COM1.”, close the tool and open the Hyperterminal to catch the serial port information.

User should open Hyperterminal and Input any key to enter U-BOOT prompts **in three seconds**, or else U-BOOT will load default parameter.

3.1.2 Update images from Ethernet

SBC8118 board can update images through Ethernet, insert TF card to update images with u-boot prompts, this section show to update image using Ethernet.

Let's assign the below IP's for the PC and the single board computer:

PC: 192.192.192.154

Single board computer: 192.192.192.215

1) PC TFTP service

a) As shown in the following Figure, launch the tftpd32.exe from the CD under the folder CD\linux\tools, and click "Browse" to set the sharing space:

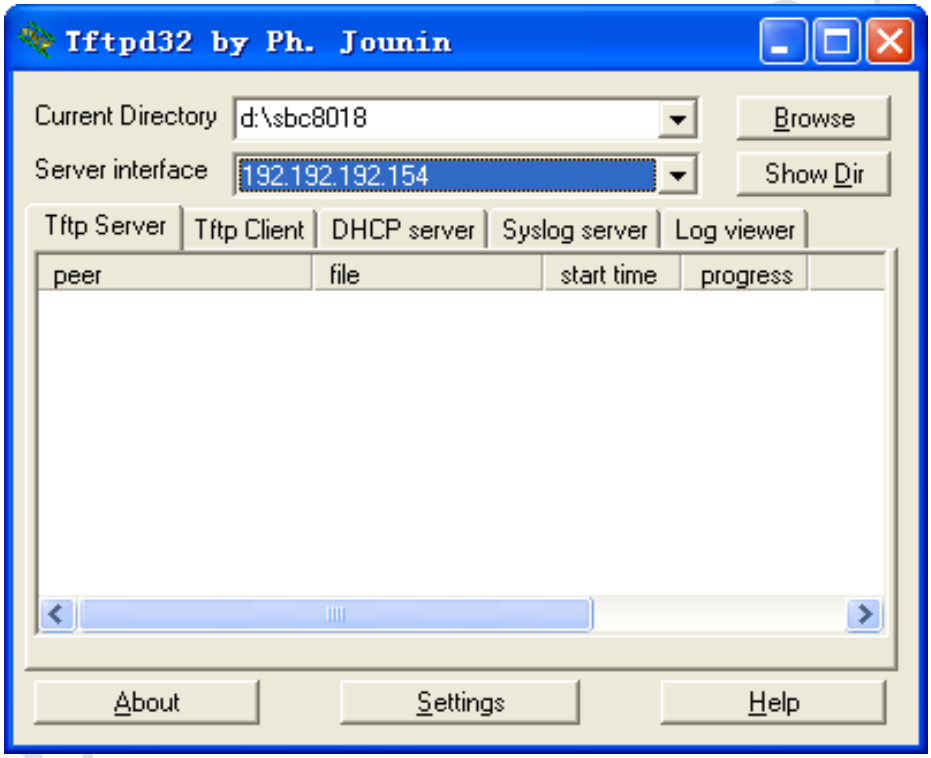


Figure 3-5 tftpd32 tool

b) Copy u-boot-nand-ais.bin, u-boot-spi-ais.bin, uImage, rootfs.img from the CD under the folder CD\linux\image\ to the folder d:\SBC8118.

2) On U-BOOT prompts input the commands as below:

a) Set the environment with "ipaddr" and "serverip":

```
U-Boot > setenv ipaddr 192.192.192.215
U-Boot > setenv serverip 192.192.192.154
```

b) Erase the NAND Flash

```
U-Boot > nand erase
```

```
NAND erase: device 0 whole chip
Skipping bad block at  0x0ff80000
Skipping bad block at  0x0ffa0000
Skipping bad block at  0x0ffc0000
Skipping bad block at  0x0ffe0000

OK
```

c) Write U-BOOT to NAND Flash

```
U-Boot > nand erase 0x20000 0xa0000

NAND erase: device 0 offset 0x20000, size 0xc0000
Erasing at 0xc0000 -- 100% complete.
OK
U-Boot > nandecc hw
HW ECC selected
U-Boot > tftp 0xc0700000 u-boot-nand-ais.bin;nand write.i 0xc0700000 0x20000 ${filesize}
Using  device
TFTP from server 192.192.192.154; our IP address is 192.192.192.215
Filename 'u-boot-nand-ais.bin'.
Load address: 0xc0700000
Loading: #################
done
Bytes transferred = 210860 (337ac hex)
NAND write: device 0 offset 0x20000, size 0x337ac
 210944 bytes written: OK
U-Boot > nandecc sw
SW ECC selected
```

d) Write U-BOOT to SPI Flash

```
U-Boot > sf probe 0
1024 KiB SST25VF080B at 0:0 is now current device
```

U-Boot > **sf erase 0 100000**

U-Boot > **tftp 0xc0700000 u-boot-spi-ais.bin; sf write c0700000 0 40000**

Using device

TFTP from server 192.192.192.154; our IP address is 192.192.192.215

Filename 'u-boot-spi-ais.bin'.

Load address: 0xc0700000

Loading: ################

done

Bytes transferred = 210860 (337ac hex)

e) Write kernel

U-Boot > **nand erase 0x200000 0x280000**

NAND erase: device 0 offset 0x200000, size 0x280000

Erasing at 0x460000 -- 100% complete.

OK

U-Boot > **nandecc sw**

SW ECC selected

U-Boot > **tftp 0xc0700000 uImage;nand write.i 0xc0700000 0x200000 ${filesize}**

Using device

TFTP from server 192.192.192.154; our IP address is 192.192.192.215

Filename 'uImage'.

Load address: 0xc0700000

Loading: ###

###

############################

done

Bytes transferred = 2299460 (231644 hex)

NAND write: device 0 offset 0x200000, size 0x231644

2299904 bytes written: OK

f) Write file system

U-Boot > **nand erase 0x600000 0x79c0000**

U-Boot > **nandecc sw**

```
SW ECC selected
U-Boot > tftp 0xc2000000 rootfs.img;nand write.i 0xc2000000 0x600000 ${filesize}
Using    device
TFTP from server 192.192.192.154; our IP address is 192.192.192.215
Filename 'rootfs.img'.
Load address: 0xc2000000
Loading: #################################################################
         ######T
#############################################################
         ###########################################################################
         ##########################################################T            #T
#########
         #####
done
Bytes transferred = 3889116 (3b57dc hex)


NAND write: device 0 offset 0x600000, size 0x3b57dc
 3889152 bytes written: OK
```

You can use one command to complete all operation which contains TFTP download and image flashing

U-Boot > **run updatesys**

At this time, flickering of LED lamp on the board indicates that update has been finished; you just need to reboot it.

3) B oot-up

a) NAND Flash boot-up

Please make sure the switch [S3] settings are selected as below:

Switch S3-1 to ON position, the rest switches should be in OFF position.

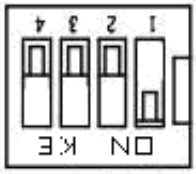


Figure 3-6 Boot-up from NAND Flash

b) SPI Flash boot-up

Make sure the toggle switch [S3] as below:

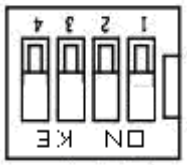


Figure 3-7 Boot-up from SPI Flash

3.2 Quick Start up with WinCE system

SBC8118 support NK.bin or NK.nb0 boot from TF Card and NAND flash, this chapter would introduce both of this system boot methods.

3.2.1 Flashing EBOOT to SPI Flash

EBOOT can update to SPI Flash with sfh_OMAP-L138.exe (.net framework require, install dotNetFx40_Full_x86_x64.exe from Microsoft).

1) Make sure serial cable had established single board computer and PC.
2) Switch S3-3 and S3-4 to ON position, the other switch to OFF position.

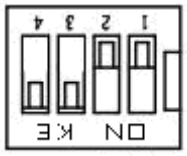


Figure 3-8

3) Copy the folder “bin” from the CD folder CD\wince6\tools to the folder D:\SBC8118.
4) According to your LCD size, copy the EBOOTSPIFLASH.nb0 from the CD folder WINCE600\Image\4_3INCH or WINCE600\image\7INCH to the folder D:\SBC8118\bin.
5) Click Start -> All Programs -> run, and input “CMD” on the pop-up dialog to enter Windows Command Prompt(cmd.exe), input the following commands as below:

```
d:
```

cd \SBC8118\bin

6) Run the flash tool to erase the SPI Flash: (change COM port if required)

sfh_OMAP-L138.exe -erase -targetType AM1808 -flashType SPI_MEM -p COM1

7) Power on the Kit. You should see progress being reported, wait until it completes, and then power off the kit.

Note: If the erase sequence does not complete after 30 seconds press a key to terminate the sfh_OMAP-L138.exe program and continue with the flashing procedure.

8) Run the flash tool to write an appropriate UBL(First Bootloader) and EBOOT to flash (change COM port if required).

sfh_OMAP-L138.exe -flash -targetType AM1808 -flashType SPI_MEM -v -p COM1 -appStartAddr 0xc7f60000 -appLoadAddr 0xc7f60000 ubl-spi-ais.bin EBOOTSPIFLASH.nb0

9) Power on the Kit. You should see progress being reported, wait until it completes.

```
Flashing UBL arm-nand-ais-456mhz.bin (13776 bytes) at 0x00000000

        Target: SENDIMG
        Target:   BEGIN
 100% [
                Image data transmitted over UART.

        Target:    DONE
 100% [
                    UBL programming complete

        Target: SENDING
        Target:    DONE

Flashing application EBOOTNANDFLASH.nb0 (262144 bytes) at 0x00010000

        Target: SENDIMG
        Target:   BEGIN
 100% [
                Image data transmitted over UART.

        Target:    DONE
 100% [
                 Application programming complete

        Target: Number of blocks needed for header and data: 0x0x00000003
        Target: Attempting to start in block number 0x0x00000006.
        Target: Magicnum: 0x0x55424CBB
        Target: Entrypoint: 0x0xC7F60000
        Target: Numpage: 0x0x00000080
        Target: Writing header and image data to Block 0x00000006, Page 0x000000
00
        Target:    DONE
        Target:    DONE

Operation completed successfully.
```

Figure 3-9

10) Power off the Kit and set DIP switches S3-1~S3-4 to OFF to select boot from SPI.

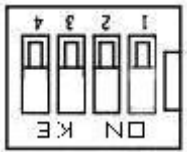


Figure 3-10

11) Start your serial terminal application (**115200 baud, 8N1**)

12) Power on the Kit and check that the new EBOOT image boot.

3.2.2 Flashing EBOOT to NAND Flash

EBOOT can update to NAND Flash with sfh_OMAP-L138.exe (.net framework require, install dotNetFx40_Full_x86_x64.exe from Microsoft).

1) Make sure serial cable had established single board computer and PC.

2) Switch S3-3 and S3-4 to ON position, the other switch to OFF position.

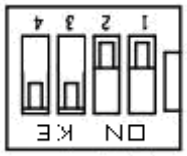


Figure 3-11

3) Copy the folder “bin” from the CD folder CD\wince6\tools to the folder D:\SBC8118.

4) According to your LCD size, copy the EBOOTNANDFLASH.nb0 from the CD folder WINCE600\Image\4_3INCH or WINCE600\image\7INCH to the folder D:\SBC8118\bin.

5) Click Start -> All Programs -> run, and input “CMD” on the pop-up dialog to enter Windows Command Prompt(cmd.exe), input the following commands as below:

```
d:
cd \SBC8118\bin
```

6) Run the flash tool to erase the NAND Flash: (change COM port if required)

```
sfh_OMAP-L138.exe -erase -targetType AM1808 -flashType NAND -p COM1
```

7) Power on the Kit. You should see progress being reported, wait until it completes, and then power off the kit. Note: If the erase sequence does not complete after 30 seconds press a key to terminate the sfh_OMAP-L138.exe program and continue with the flashing procedure.

8) Run the flash tool to write an appropriate UBL(First Bootloader) and EBOOT to flash (change COM port if required).

```
sfh_OMAP-L138.exe -flash -targetType AM1808 -flashType NAND -v -p COM1
-appStartAddr 0xc7f60000 -appLoadAddr 0xc7f60000 ubl-nand-ais.bin
```

EBOOTNANDFLASH.nb0

9) Power on the Kit. You should see progress being reported, wait until it completes.

```
Flashing UBL arm-nand-ais-456mhz.bin (13776 bytes) at 0x00000000

        Target: SENDIMG
        Target:   BEGIN
 100% [
                  Image data transmitted over UART.

        Target:    DONE
 100% [
                      UBL programming complete

        Target: SENDING
        Target:    DONE

Flashing application EBOOTNANDFLASH.nb0 (262144 bytes) at 0x00010000

        Target: SENDIMG
        Target:   BEGIN
 100% [
                  Image data transmitted over UART.

        Target:    DONE
 100% [
                   Application programming complete

        Target: Number of blocks needed for header and data: 0x0x00000003
        Target: Attempting to start in block number 0x0x00000006.
        Target: Magicnum: 0x0x55424CBB
        Target: Entrypoint: 0x0xC7F60000
        Target: Numpage: 0x0x00000080
        Target: Writing header and image data to Block 0x00000006, Page 0x000000
00
        Target:    DONE
        Target:    DONE

Operation completed successfully.
```

Figure 3-12

10) Power off the Kit and set DIP switches S3-1 to ON, all others to OFF.

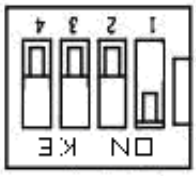


Figure 3-13

11) Start your serial terminal application (115200 baud, 8N1)

12) Power on the Kit and check that the new EBOOT image boot.

3.2.3 Update TF Card NK runtime images

1) Format TF card

Format the TF Card in FAT/FAT32 file system.

2) Copy NK runtime image

a) Navigate to the directory WINCE600/image/lcd7inch or WINCE600/image/lcd4.3inch, according to the LCD size.

b) Copy NK.nb0/NK.bin to TF card.

3) Change the EBOOT settings to boot NK from TF Card

Insert the TF card to kit, power on the kit and press space button to enter the EBOOT menu.

a) Press the key [2] -> [2] -> [2] by step to select boot NK from TF card

```
Booting with TI UBL
Device OPP (456MHz, 1.3V)01
Microsoft Windows CE Bootloader Common Library Version 1.4 Built Sep 23 2011 15:29:43
INFO:OALLogSetZones: dpCurSettings.ulZoneMask: 0xb
Microsoft Windows CE EBOOT 1.0 for AM1808 OMAPL138/AM18X EVM. Built Sep 23 2011 at 15:30:38
BSP version 1.3.0, SOC version 1.3.0
    CODE : 0xC7F60000 -> 0xC7FA0000
    DATA : 0xC7FA0000 -> 0xC7FE0000
   STACK : 0xC7FE0000 -> 0xC8000000
Enabled OAL Log Zones :    ERROR, WARN, INFO,
Platform Init done
System ready!
Preparing for download...
Predownload...
FMD: ReadID (Mfg=0x2c, Dev=0xda)
WARN: Invalid boot configuration found (using defaults)
Lan MAC: 00:08:ee:00:00:00
INFO: MAC address: 00:08:ee:00:00:00
WARN: Invalid BSP_ARGS data found (using defaults)
WARN: Unable to get hardware entropy

Hit space to enter configuration menu    2

--------------------------------------------------------------------------------
```

Main Menu

---

[1] Show Current Settings

[2] Boot Settings

[3] Network Settings

[5] Video Settings

[6] Save Settings

[7] Peripheral Tests

[R] Reset Settings To Default Values

[0] Exit and Continue

Selection: **2**

Boot Settings

---

[1] Show Current Settings

[2] Select Boot Device

[3] Select Boot Delay

[4] Select Debug Device

[5] Force Clean Boot

[6] Write Download RAM NK to Flash

[7] Set Device ID String

[8] Allow DSP to Boot

[0] Exit and Continue

Selection: **2**

Select Boot Device

---

[1] EMAC

[2] NK from SD

[3] NK from NAND flash

[0] Exit and Continue

Selection (actual NK from SD): **2**

Boot device set to NK from SD

b) Press the key [0] -> [0] by step to start system from SD card, and you would see the following message in serial terminal:

Boot Settings

---

[1] Show Current Settings

[2] Select Boot Device

[3] Select Boot Delay

[4] Select Debug Device

[5] Force Clean Boot

[6] Write Download RAM NK to Flash

[7] Set Device ID String

[8] Allow DSP to Boot

[0] Exit and Continue

Selection: **0**

Main Menu

---

[1] Show Current Settings

[2] Boot Settings

[3] Network Settings

[5] Video Settings

[6] Save Settings

[7] Peripheral Tests

[R] Reset Settings To Default Values

[0] Exit and Continue

Selection: **0**

Device ID set to AM1808-0

BLFlashDownload: LogicalLoc - 0x01C40000

Loading from SD card

+ReadNKFromSDMMC

ReadFileFromSDMMC: reading file 'nk.bin'

SDBootPDD: PDD_SDInitializeHardware: MMCSD

SDBootMDD: SDInitializeHardware: SD card detected

SDBootMDD: SDInitializeHardware: V2.0 card detected

SDBootMDD: SDInitializeHardware: timeOut = 0

SDBootMDD: SDInitializeHardware: timeOut = 1

SDBootMDD: SDInitializeHardware: timeOut = 2

SDBootMDD: SDInitializeHardware: timeOut = 3

SDBootMDD: Card address is 1234

SDBootMDD: 4-bit data bus selected

InitMasterBootRecord: Partition 0, type 12

InitMasterBootRecord: Partition 0, FAT32, start 0x7e00, length 0x753f8200

InitPartition: Offset 0x7e00, length 0x753f8200

ReadFileFromSDMMC: file size = 16138467 bytes

UnpackBINImage: unpacking binary from 0xc2000000

UnpackBINImage: Image start = 0x80000000

UnpackBINImage: Image length = 0x102fd2c

UnpackBINImage: record 0, start=0x80000000, length=0x4, checksum=0x1eb

…….

UnpackBINImage: record 296, start=0x0, length=0x80001000, checksum=0x0

CheckCEImage: checking image at 0xc0000000

```
ROMHDR (pTOC = 0xc102de3c) ----------------------
   DLL First                : 0x4001c001
   DLL Last                 : 0x40b5c097
   Physical First           : 0x80000000
   Physical Last            : 0x8102fd2c
   Num Modules              :          181
   RAM Start                : 0x81030000
   RAM Free                 : 0x8103f000
   RAM End                  : 0x8373f800
   Num Copy Entries         :            2
   Copy Entries Offset : 0x804f4fd4
   Prof Symbol Length    : 0x00000000
   Prof Symbol Offset    : 0x00000000
   Num Files                :           73
   Kernel Flags             : 0x00000000
   FileSys RAM Percent : 0x30303030
   Driver Glob Start     : 0x00000000
   Driver Glob Length    : 0x00000000
   CPU                      :       0x01c2
   MiscFlags                :       0x0002
   Extensions               : 0x80001070
   Tracking Mem Start    : 0x00000000
   Tracking Mem Length : 0x00000000
-----------------------------------------------

Image Start .......: 0x00000000
Image Size ........: 0x00000000
Image Launch Addr .: 0x00000000
Image ROMHDR ......: 0x00000000
Boot Device/Type ..: 3 / 6
```

```
ADEhellounch Windows Embedded CE by jumping to 0xc0000000...
Windows CE Kernel for ARM (Thumb Enabled) Built on Oct 20 2009 at 18:39:19
OEMInit: init.c built on Sep 28 2011 at 15:51:27.
BSP version 1.3.0, SOC version 1.3.0
INFO:OALLogSetZones: dpCurSettings.ulZoneMask: 0xf
WARN: Updating local copy of BSP_ARGS
Intr Init done...
Timer Init done...
+OALDumpClocks
Clock Configuration :
  Reference Clock 0 ..  24000000 Hz
    PLL0 ..... 456000000 Hz
    PLL0:SYSCLK1 ..... 456000000 Hz (DSP Subsystem)
    PLL0:SYSCLK2 ..... 228000000 Hz (UART,EDMA,SPI,MMC/SD,VPIF,LCDC,SATA,uPP,USB2.0,HPI,PRU)
    PLL0:SYSCLK3 .....  91200000 Hz (EMIFA)
    PLL0:SYSCLK4 ..... 114000000 Hz (INTC, SYSCFG, GPIO, PSC, I2C1, USB1.1, EMAC/MDIO, GPIO)
    PLL0:SYSCLK5 ..... 152000000 Hz (reserved)
    PLL0:SYSCLK6 ..... 456000000 Hz (ARM Subsystem)
    PLL0:SYSCLK7 .....  76000000 Hz (EMAC)
    PLL0:AUXCLK ......  24000000 Hz (I2C0, Timers, McASP0 serial clock, RTC, USB2.0 PHY)
    PLL1 ..... 264000000 Hz
    PLL1:SYSCLK1 ..... 264000000 Hz (DDR2/mDDR PHY)
    PLL1:SYSCLK2 ..... 132000000 Hz (Optional for: McASP0,McBSP,ePWM,eCAP,SPI1)
    PLL1:SYSCLK3 .....  88000000 Hz (PLL0 input)
-OALDumpClocks
-OEMInit
  PINMUX14=0x00000000
  PINMUX15=0x00000000
```

```
    PINMUX16=0x22222200
    PINMUX17=0x22222222
    PINMUX18=0x22000022
    PINMUX19=0x02000022
OEMGetExtensionDRAM: Added 0x84400000 -> 0x88000000
OEM: Cleaning system hive
OEM: Cleaning user profiles
WARN: Updating local copy of BSP_ARGS
OEM: Not cleaning system hive
FMD: ReadID (Mfg=0x2c, Dev=0xda)
MICBIASHardwareContext::Init 555
Adapter's MAC address is 00:08:EE:00:00:00
SDHC +Init
        SDHC Active RegPath: Drivers\Active\21
                                        +SDHCPDD_Init: Ctrl 0, Entry
SDHC -Init
        SDHC +Open
                SDHC +Open
                        SDHC_CARD_DETECT = 1
SDHC CommandCompleteHandler: Command response timeout
SDHC CommandCompleteHandler: Command response timeout
SDHC CommandCompleteHandler: Command response timeout
SDHC CommandCompleteHandler: Command response timeout
SDHC CommandCompleteHandler: Command response timeout
SDHC CommandCompleteHandler: Command response timeout
SDHC CommandCompleteHandler: Command response timeout
SDHC CommandCompleteHandler: Command response timeout
```

3.2.4 Flashing NK.bin to NAND flash

This section introduce how to flashing NK.bin to nand flash via ethernet and VS2005:

1) Confirm you have performed the release build in C:\WINCE600\OSDesigns\SBC8118\ SBC8118.sln.
2) Connect PC and SBC8118 with RJ45 Cable.
3) Switch the S3-1 to ON position, the other bits switch to OFF position:

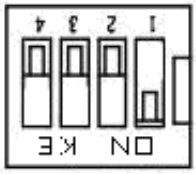


Figure 3-14 Boot up from NAND Flash

4) Select EMAC as boot media in eboot menu, press key [2]->[2]->[1] , steps as below:

```
---------------------------------------------------------------------------
Select Boot Device
---------------------------------------------------------------------------
 [1] EMAC
 [2] NK from SD
 [3] NK from NAND flash
 [0] Exit and Continue

 Selection (actual NK from SD): 1
 Boot device set to EMAC
```

5) Press [6] -> [y] enable flashing NK to NAND Flash in EBOOT menu:

```
---------------------------------------------------------------------------
 Boot Settings
---------------------------------------------------------------------------
 [1] Show Current Settings
 [2] Select Boot Device
 [3] Select Boot Delay
 [4] Select Debug Device
 [5] Force Clean Boot
```

[6] Write Download RAM NK to Flash

[7] Set Device ID String

[8] Allow DSP to Boot

[0] Exit and Continue

Selection: **6**

Enable Write Download RAM NK to Flash (actually disabled) [y/-]: **y**

Write Download RAM NK to Flash enabled

6) Press key [0] return to eboot main menu, press key [3] to set the network property. Setting the DHCP, IP, NETMASK according to your network environment, confirm that the IP, NETMASK of kit is in the same sub network of your PC.

Network Settings

---

[1] Show Current Settings

[2] Enable/disable KITL

[3] KITL interrupt/poll mode

[4] Enable/disable DHCP

[5] Set IP address

[6] Set IP mask

[7] Set default router

[8] Enable/disable VMINI

[0] Exit and Continue

Selection: **1**

Network:

KITL state ..... disabled

KITL mode ..... interrupt

DHCP ..... disabled

MAC address 00:08:ee:00:00:00

IP address 192.192.192.200

IP mask 255.255.255.0

IP router 192.192.192.101

```
VMINI ..... disabled
```

7) Press key [0] return to EBOOT main menu, then press key [0] prepare to download NK.bin from PC, and the serial terminal would display the message as below:

```
INFO: Boot device uses MAC 00:08:ee:ff:ff:ff
+EbootSendBootmeAndWaitForTftp
Sent BOOTME to 255.255.255.255
Sent BOOTME to 255.255.255.255
Sent BOOTME to 255.255.255.255
Sent BOOTME to 255.255.255.255
Sent BOOTME to 255.255.255.255
Sent BOOTME to 255.255.255.255
Sent BOOTME to 255.255.255.255
```

8) Click [Target->Connectivity Options] in VS2005 Menu, and then pop up the dialog below, Select Ethernet in Download Item select list.

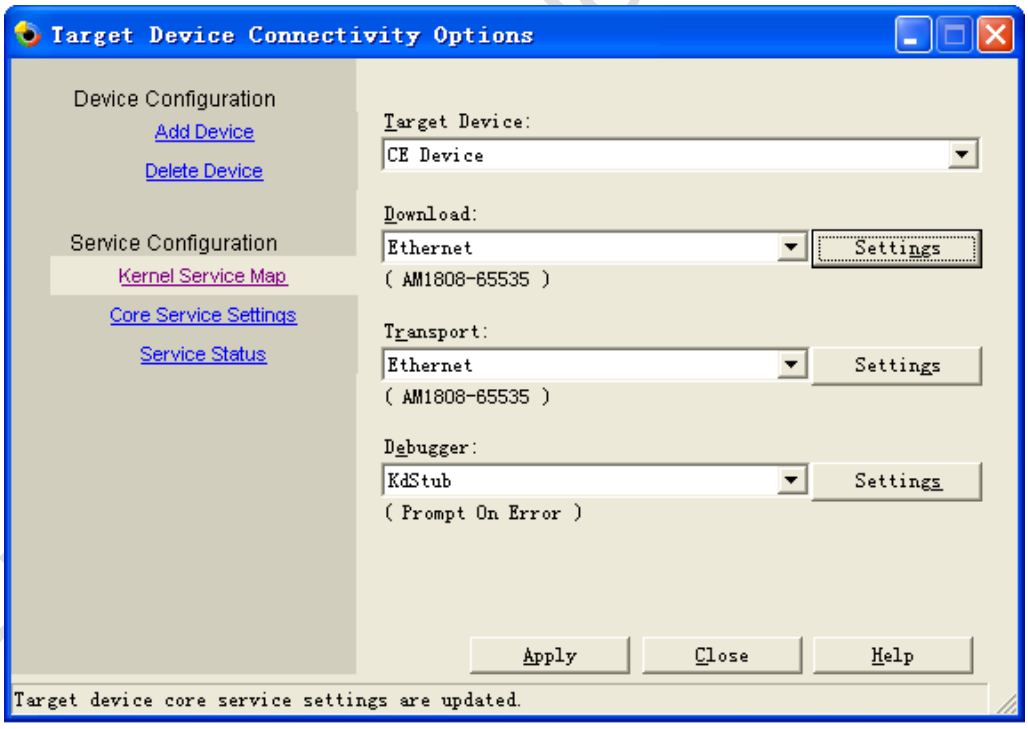


Figure 3-15

9) Click Settings button locate in the right of Download Item select list, then it would pop up a dialog as picture below, if the network properly setting and network cable is OK, you would see device AM1808-65535 in Active target devices edit box, select device AM1808-65535 and click [OK] button back to Connectivity Options setting dialog, click

[Apply->Close->finish] Connectivity Options setting.

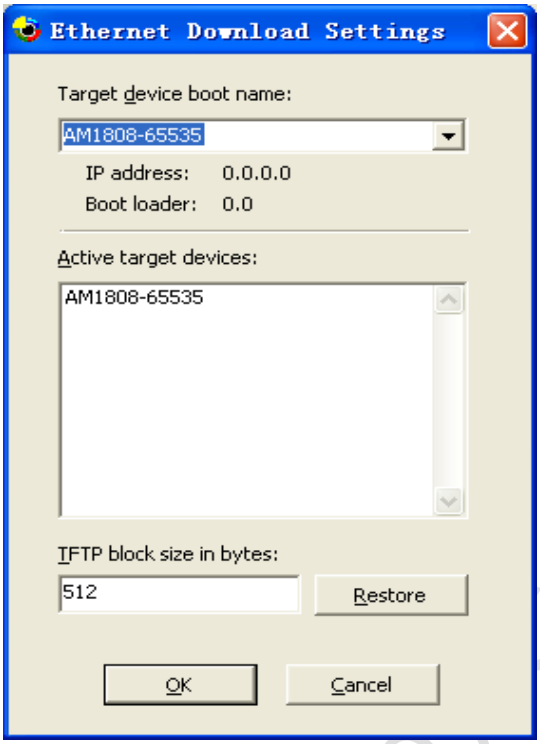


Figure 3-16

10) Click [Target] -> [Attach Device] in VS2005 menu Start download NK.bin, then VS2005 would pop up a dialog to indicate download progress, after download and NK.bin flashing finish the serial terminal would display the message below:

```
OEMWriteFlash: NK written
ROMHDR at Address 80000044h
Image Start .......: 0x80000000
Image Size ........: 0x00ff9a74
Image Launch Addr .: 0x80001000
Image ROMHDR ......: 0xc0ff7be0
Boot Device/Type ..: 2 / 6
Got EDBG_CMD_JUMPIMG
Got EDBG_CMD_CONFIG, flags:0x00000000
BLFlashDownload: LogicalLoc   - 0x62000000
Load NK image from flash memory (NAND)
FMD: ReadID (Mfg=0x2c, Dev=0xda)
BLFlashDownload: cp1
```

power down and power on the kit, press space button to eboot menu, press key [2]->[2]->[3] by

step to select boot NK from NAND flash, press [0] -> [0] to start system from NAND flash.

Technical support & Warranty Service

Embest Technology Co., Ltd., established in March of 2000, is a global provider of embedded hardware and software. Embest aims to help customers reduce time to market with improved quality by providing the most effective total solutions for the embedded industry. In the rapidly growing market of high end embedded systems, Embest provides comprehensive services to specify, develop and produce products and help customers to implement innovative technology and product features. Progressing from prototyping to the final product within a short time frame and thus shorten the time to market, and to achieve the lowest production costs possible. Embest insists on a simple business model: to offer customers high-performance, low-cost products with best quality and service. The content below is the matters need attention for our products technical support and warranty service:

Technical support service

Embest provides one year free technical support service for all products. Technical support service covers:

- Embest embedded platform products software/hardware materials
- Assist customers compile and run the source code we offer.
- Solve the problems accurs on embeded software/hardware platform if users follow the instructions in the documentation we offer.
- Judge whether the product failure exists.

Special explanation, the situations listed below are not included in the range of our free technical support service, and Embest will handle the situation with discretion:

- Software/Hardware issues user meet during the self-develop process
- Issues happen when users compile/run the embedded OS which is tailored by users themselves.
- User's own applications.

- Problems happen during the modification of our software source code

Maintenance service clause

1) The products except LCD, which are not used properly, will take the warranty since the day of the sale:

   PCB: Provide 12 months free maintenance service.

2) The situations listed below are not included in the range of our free maintenance service, Embest will charge the service fees with discretion:

   A. Can't provide valid Proof-of-Purchase, the identification label is torn up or illegible, the identification label is altered or doesn't accord with the actual products;
   B. Don't follow the instruction of the manual in order to damage the product;
   C. Due to the natural disasters (unexpected matters), or natural attrition of the components, or unexpected matters leads to the defects of appearance/function;
   D. Due to the power supply, bump, leaking of the roof, pets, moist, impurities into the boards, all those reasons which lead the defects of appearance/function;
   E. User unauthorized weld or dismantle parts leads the product's bad condition, or let other people or institution which are not authorized by Embest to dismantle, repair, change the product leads the product bad connection or defects of appearance/function;
   F. User unauthorized install the software, system or incorrect configuration or computer virus leads the defects;
   G. Purchase the products through unauthorized channel;
   H. Those commitments which is committed by other institutions should be responsible by the institutions, Embest has nothing to do with that;

3) During the warranty period, the delivery fee which delivery to Embest should be covered by user, Embest will pay for the return delivery fee to users when the product is repaired. If the warranty period is expired, all the delivery fees will be charged by users.

4) When the board needs repair, please contact technical support department.

Note: Those products are returned without the permission of our technician, we will not take any responsibility for them.

Basic notice to protect and maintenance LCD

1) Do not use finger nails or hard sharp object to touch the surface of the LCD, otherwise user can't enjoy the above service.
2) Embest recommend user to purchase a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.
3) Do not clean the surface of the screen with chemicals, otherwise user can not enjoy above service.

Note: Embest do not supply maintenance service to LCDs. We suggest the customer first check the LCD after getting the goods. In case the LCD cannot run or show no display, customer should inform Embest within 7 business days from the moment of getting the goods.

Value Added Services

We will provide following value added services:

- Provided services of driver develop based on Embest embedded platform, like serial port, USB interface devices, LCD screen.
- Provided the services of control system transplant, BSP drivers develop, API software develop.
- Other value added services like power adapter, LCD parts.
- Other OEM/ODM services.
- Technically training.

Please contact Embest to get technical support:

- Support Tel:+86-755-25635626-872/875/897
- Fax:+86-755-25635626-666

- Pre-Sale consultation: market@embedinfo.com
- After-Sale consultation: support@embedinfo.com