# SBC6045 User Manual

Rev. 1.2

Release: 2013-02-26

*Thank you for purchasing our products, we will do better because of you.*

**Embest Info&Tech Co., LTD.**

Room 509, Luohu Science&Technology Building,
#85 Taining Rd., Shenzhen, Guangdong, China 518020
Tel: +86-755-25635656/25636285
Fax: +86-755-25616057
Email: market@embedinfo.com
http://www.embedinfo.com/english
http://www.armkits.com

# Revision record

| Date | Revision Version | CR ID/Defect ID | Section Number | Change Description | Author |
|---|---|---|---|---|---|
| 2011-05-27 | 1.0 | | | | |
| 2011-08-02 | 1.1 | | | | |
| | | | | | |



Attention

As the product version upgrades or other reasons, the contents of this document will be updated from time to time. Unless otherwise agreed, the use of this document only as a guide, all statements in this document, information and suggestions do not constitute any express or implied.

# Table of Contents

# Part 1 Overview

## Chapter 1 System Overview

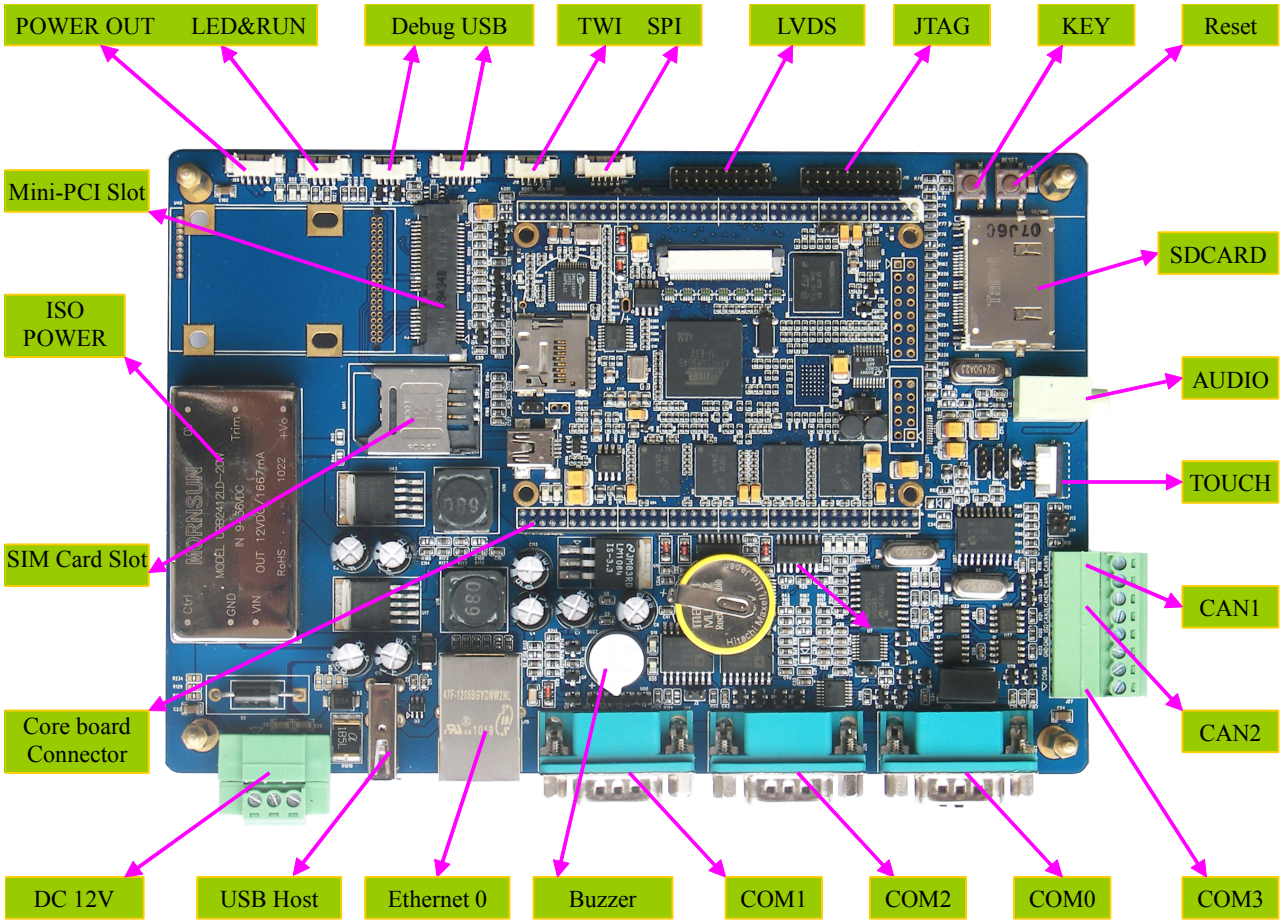
Embest SBC6045 is a high-performance ARM embedded single board computer (SBC) provided versatile communication interfaces to meet a wide variety of applications such as industrial control, automation, equipment monitoring, data logging, medical equipment, embedded web server etc. It is based on the 400MHz Atmel AT91SAM9G45 microprocessor and features wide input range from 12V to 36V with isolated power.

The SBC6045 board uses Mini6045 processor card as the CPU core board which has 256MB (2*128MByte) DDR2 SDRAM, 256MB Nand Flash, 4MB Data Flash, 2Kbit EEPROM and integrates UART, Ethernet, USB OTG, LCD, Touch screen, RTC and TF card slot on board. It connects to the expansion board through two 2.0mm pitch 80-pin connectors which enables the expansion board to access to all the core board's I/O through on-board connectors. The SBC6045 also has expansion blocks such as CAN and SD card slot as well as a Mini-PCI slot and SIM card slot which has integrated UART and USB signals. Customer can attach 3G module, GPRS module and other Mini-PCI devices to this interface.

The board is capable of supporting Linux 2.6.30 open source operating system. Embest provides BSP package, user manual and some other tools and documents to help customer with their development. It would be a ideal platform for your evaluation and fast your development work. It can be also directly used for your design

# Part 2 Hardware system

## Chapter 2 Specifications of hardware




SBC6045 consists of three parts：

- Main board
- core board
- LCD module 4.3inch 480*272（optional）
- LCD module 7.0inch 800*480（optional）

### 2.1 CPU

Pocessor:T91SAM9G45，ARM926EJ-STM ARM

AT91SAM9G45 chip applies an ARM926EJ-S core with MMU function, which comprises a 64KB internal SRAM and a 64KB internal ROM,2 external bus interfaces. It can coordinate 4 Pieces DDR2/LPDDR,or 4 Pieces SDRAM/LPSDR，otherwise 4 Static Memory,or in the term of the same

number of CF Flash Memories, SLC NAND Flash with ECC checking ability.
AT91SAM9G45 combinates the function of User Interface and High Speed Data Connection together, such as LCD controller, Resistance Touch Screen, Camera input, Audio input, 10/100M Ethernet,High speed USB, SDIO and so on. AT91SAM9G45 provides user with outstanding experience through outstanding Internet and local memory media,operating at 400MHz and a number of external devices working at more than 100Mbps.
AT91SAM9G45 play the latest DDR2 and NAND flash memory interface for program and data storage. With a 133M internal multilayer bus interface of 37 DMA channels relative, a dual external bus interface,and a 64K byte distributed EMS memory which can deploy TCM, the necessary bandwidth for maintaining the communication between Processor and High spped external device is achievable.
AT91SAM9G45 efficient power management controller clock gating and battery backup part,
AT91SAM9G45 power management controller in the power and standby mode reduces the power consumption to a minimum.

## 2.2 Core board functions

| CLASS | FUNCTION | PARAMETER |
|---|---|---|
| Pocessor | CPU | Atmel AT91SAM9G45, ARM9,400MHz |
| Memory | DDR2 | 64MB*2*2 |
| | NandFlash | 256MB |
| | DataFlash | 4MB |
| | EEPROM | 2Kb |
| | TF CARD | A TF card interface |
| LCD | LCD | FPC connector leads out 3 output,in a 24-bit LCD interface,touch screen interface,And the LCD interface,LVDS interface,leads |
| Others | LED | An LED system status indicator, the other is for the POWER status indicator |
| | Extended RTC | Extending an accurate RTC via the DS3231 |
| | Network | System produces a 100M network interface |
| | USB | Extending a USB 2.0 OTG interfaces through the MINI USB connector |
| | UART | Output RS232 voltage, leads out DEBUG interface |
| | Other IOs | Other interfaces come out through 160PIN 2.0mm double straight pin |

Core board size：67*82mm
The main connector : (80PIN double Straight pin) * 2 2.0mm interval
The specific pin definition please refer to ： System Electrical Description

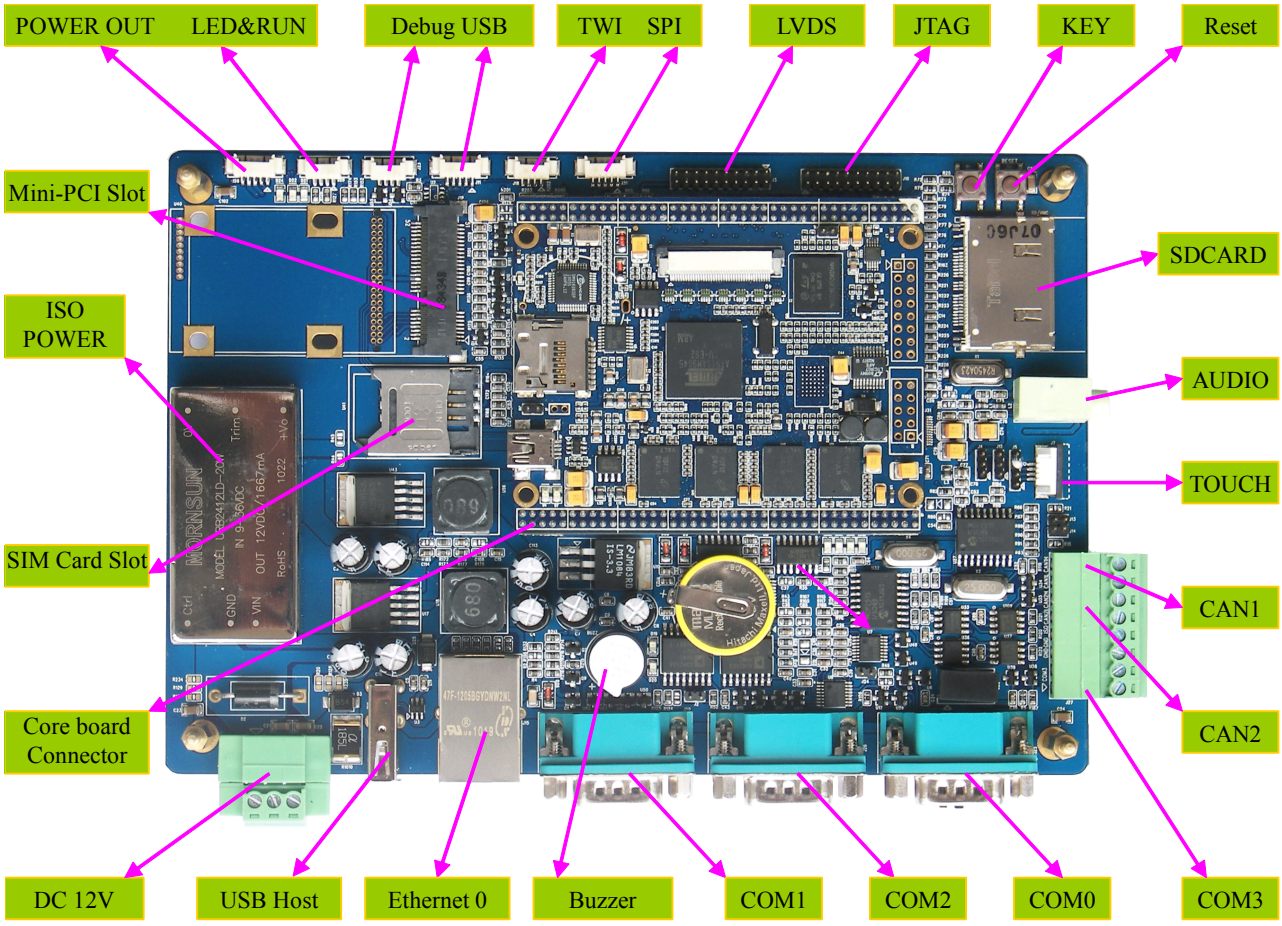
## 2.3 Functional interfaces of main board

| CLASS | FUNCTION | PARAMETER |
|---|---|---|
| Storage | SD card | High-Speed SD Card interface supports hot-swappable |
| Power | Isolated powe | Wide voltage range of 12 ~ 30V, 12V power input defaultly |
| | | DEBUG：RS232 3 wire |

SBC6045 Serial port Communication

| | | |
|---|---|---|
| | | COM0：RS232/RS485 (Isolated) 3- lines |
| | | COM1：RS232 5 wire |
| | | COM2：RS232/RS485 (Isolated) 3 - lines |
| | | COM3：RS232/TTL 3 wire |
| | CAN | 2 CAN2.0 communication interfaces get power and isolate signal by two SPI expansion |
| | NET | MCU internal built-in controller, 10/100Mbps port with indicator |
| | USB Host | 1xUSB2.0 Host controller supports USB keyboard, USB mouse, etc.. |
| | USB | 1x USB2.0 OTG controller |
| | TWI1 | Maximum 400Kbps, to support master-slave mode, comes out in the form of wire connector |
| | SPI1 | Wiring connector mode leads out |
| LCD | LCD Interface | CPU supports a maximum resolution of 1280 * 860 pixel, 24/16 bit RGB color mode, the current 4.3,7-inch screen matching well, supports 24-bit LVDS interface output |
| Input /Output | TOUCH | 4-lines resistance touch screen interface |
| | Audio Out | Support MP3 Play |
| Panel | Buzzer | A controled BUZZER by GPIO |
| | JTAG | Standard JTAG Interface 20PIN 2.0mm |
| | KEY | A GPIO button, a RESET button |
| | Indicator | Wiring connector leads to power and system status IO, useful for indication of system status on shell |
| | LED | Onboard power and IO status indicator |
| Others | RTC | Onboard rechargeable coin cell power supply for RTC |
| | Special Power Supply | Wiring connector leads to 12V power supply output, switch, backlight and power supply management |

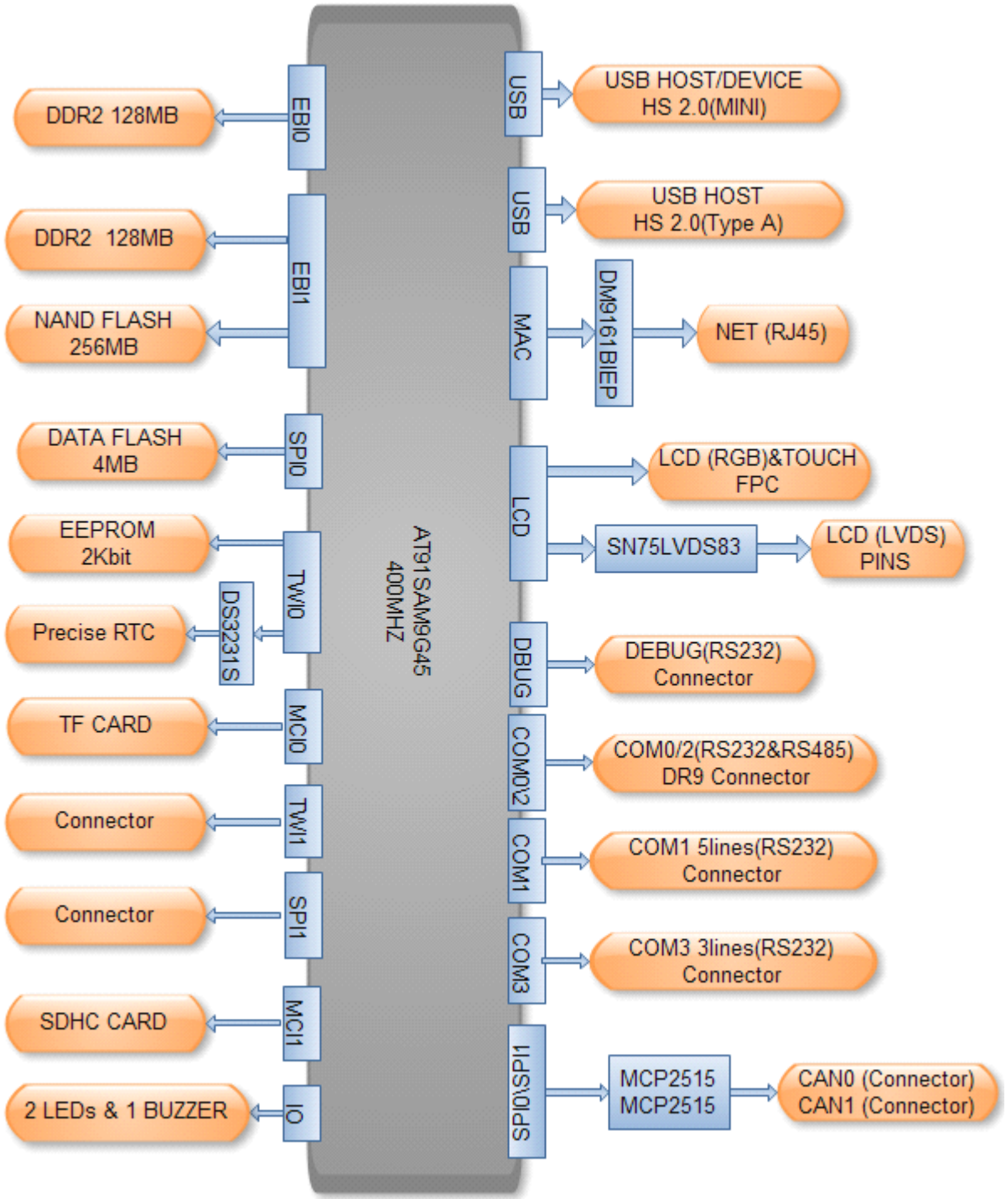
Main board picture：




Dimensions of PCB：185*125mm

Connector pin definitions from board bottom to the core board, please refer to： System Electrical Description

2.4 General system structure diagram




# Chapter 3 Description of interfaces

## 3.1 LAYOUT

SYSTEM LAYOUT DIAGRAM:：

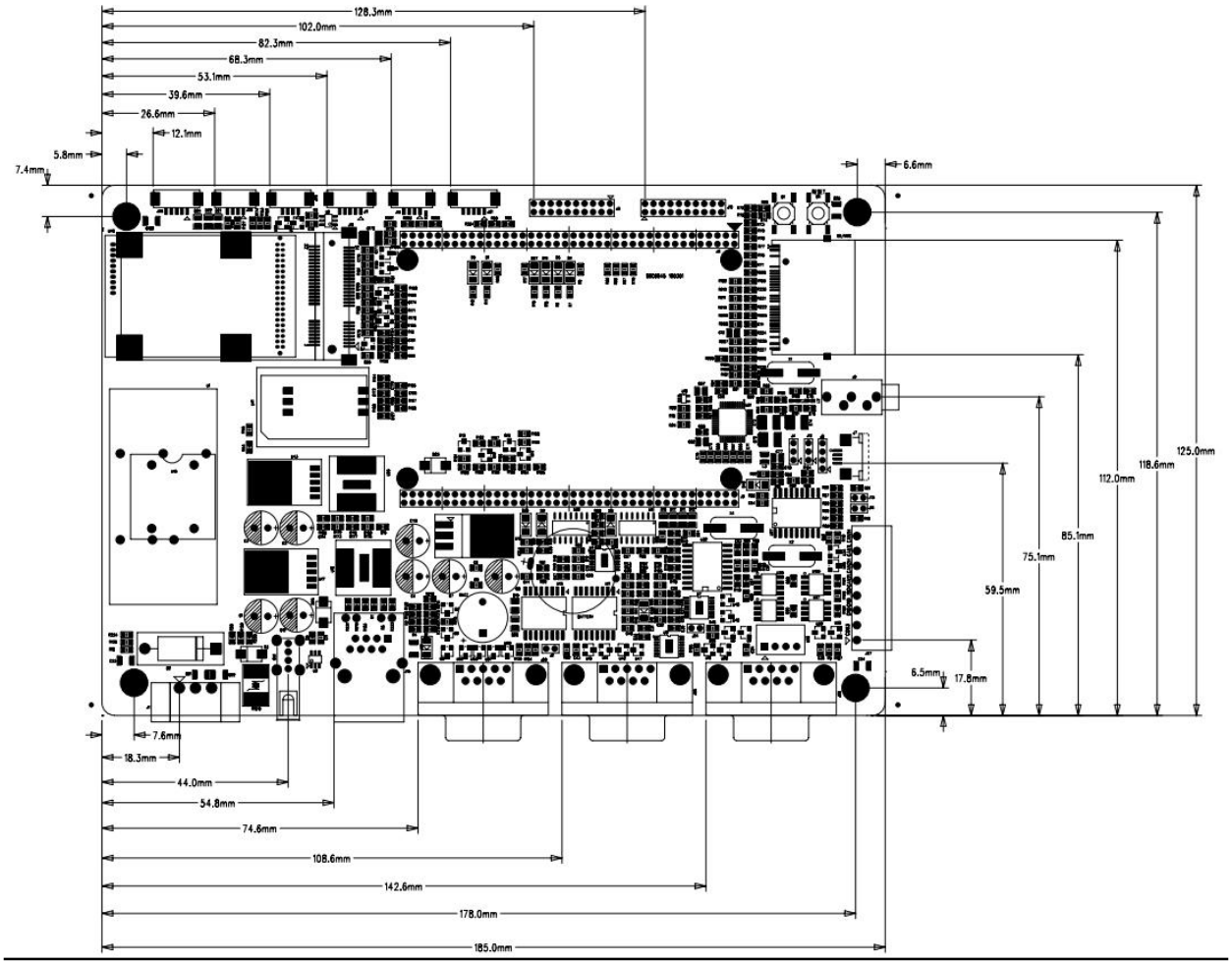



Table 2-1 Interface Details

## 3.2 POWER（J1）

- Standard 3.81mm Wiring connector can be used to provide the power input +12 V
- Input voltage range :12-36V
- Isolated maximum power limit: 20W
- The power can be protected by fuse, 1.5A limited
- Interface pins are defined as follows: (FGND for the chassis ground)

| | |
|---|---|
| VCC | 1 |
| GND | 2 |
| FGND | 3 |

Power input interface functional definition

## 3.3 SDHC（SD/MMC）

- High-speed SD card interface
- Compatible SD / MMC interface card
- 8-bit parallel bus data transfer mode
- SD / MMC interface reference CPU resources as follows:

| | | |
|---|---|---|
| SDHC | PA22 | MMC_CDA |
| | PA23 | MMC_DA0 |
| | PA24 | MMC_DA1 |
| | PA25 | MMC_DA2 |
| | PA26 | MMC_DA3 |
| | PA27 | MMC_DA4 |
| | PA28 | MMC_DA5 |
| | PA29 | MMC_DA6 |
| | PA30 | MMC_DA7 |
| | PA31 | MMC_CK |
| | PB29 | CD |
| | PB30 | WP |

## 3.4 NET (J15)

- the CPU built-in controller
- Work speed: 100Mbps
- Work mode: RMII
- Output interface adopts RJ45 connectors
- CPU resources as follows:

| | | |
|---|---|---|
| NET | PA10 | EXT0 |
| | PA11 | EXT1 |
| | PA12 | ERX0 |
| | PA13 | ERX1 |
| | PA14 | ETXEN |
| | PA15 | ERXDV |
| | PA16 | ERXER |
| | PA17 | E_TXCK |
| | PA18 | EMDC |
| | PA19 | EMDIO |
| | PD5 | MDINT |

## 3.5 UART

UART (DEBUG, COM0, COM1, COM2, COM3), all comes from the MCU resource controller

### 3.5.1 DEBUG serial port (J23)

- Serial DEBUG connector leads with 1132R
- This interface uses 3-wire
- Resources

| | | |
|---|---|---|
| DEBUG | PB12 | DRXD |
| | PB13 | DTXD |

DEBUG corresponds to DEBUG serial port from CPU (AT91SAM9G45) .

J23 Pins definition :

| PIN | Function |
|---|---|
| 1 | DRRXD |
| 2 | DRTXD |
| 3 | GND |
| 4 | NC |

- Signal input/output level: RS232
- Data rate: 115.2kbps
- Flow Control: None

### 3.5.2 COM 0, 2 (J25, J26)

- COM 0,2 with DR9 male serial connector leads RS232 & RS485 level
- COM 0,2 are 3-lines serial port
- This feature provides directly from the CPU pins
- Resources：

| | | |
|---|---|---|
| UART0 | PB19 | TXD0 |
| | PB18 | RXD0 |
| | PB17 | RTS0 |
| UART2 | PB6 | TXD2 |
| | PB7 | RXD2 |
| | PC9 | RTS2 |

COM0、COM2 are a 5-wire universal serial port, it corresponds to USART0 and USART2 from CPU (AT91SAM9G45) .

- Signal input/output level: RS232
- Data rate: 115.2kbps
- Flow Control: None
- Connector: DR9 male

At the same time the two ports also are used for RS485 functional interfaces, data from the CPU is sent out to two functional outputs simultaneously: meanwhile RS232 and RS485, the data from RS232 and RS485 interfaces can be sent back to the CPU.

DR9 interface connector (J25,J26) Pin is defined as follow：

| | | | | | |
|---|---|---|---|---|---|
| COM0 | 1 | | COM2 | 1 | |
| | 2 | RRXD0 | | 2 | RRXD2 |
| | 3 | RTXD0 | | 3 | RTXD2 |
| | 4 | | | 4 | |
| | 5 | GND | | 5 | GND |

| | | | | | |
|---|---|---|---|---|---|
| (J25) | 6 | | (J26) | 6 | |
| | 7 | 485A1 | | 7 | 485A2 |
| | 8 | 485B1 | | 8 | 485B2 |
| | 9 | GND_ISO | | 9 | GND_ISO |

### 3.5.3 COM1（J20）

- COM1 adopts DR9 with RS232 level
- COM1 is 5- lines serial port
- This function provides directly from the CPU pins
- resources：

| | | |
|---|---|---|
| COM1 | PB4 | TXD1 |
| | PB5 | RXD1 |
| | PD16 | RTS1 |
| | PD17 | CTS1 |

COM1 is a 5-wire universal serial port, it corresponds to USART1 on MCU.

- Signal input/output level: RS232
- Data rate: 115.2kbps
- Connector: DR9 Male
- DR9 interface connector (J20) Pin is defined as follow：

| | | |
|---|---|---|
| COM1 (J20) | 1 | |
| | 2 | RRXD1 |
| | 3 | RTXD1 |
| | 4 | |
| | 5 | GND |
| | 6 | |
| | 7 | RRTS1 |
| | 8 | RCTS1 |
| | 9 | |

### 3.5.4 COM3（J27）

- COM3 adopts wiring connector with RS232 level
- COM3 is 3- lines serial port
- The function operates with the J40 connector, if use the GPRS or CDMA modules, extending J27 serial function and GPRS functions against J27 can not work at the same time.they just can work individually under the system.
- This function provides directly from the CPU pins
- resources：

| | | |
|---|---|---|
| COM3 | PB8 | TXD3 |
| | PB9 | RXD3 |

COM3 is a 3-wire universal serial port, it corresponds to USART3 on MCU.

- Signal input/output level: RS232
- Maximum data rate: 115.2kbps
- Connector: Connector（J27/U40）

Output interface (J27) pin is defined as follow：

| | | |
|---|---|---|
| COM3 | 1 | RRXD3 |
| | 2 | RTXD3 |
| | 3 | GND |
| CAN2 | 4 | GND_ISO |
| | 5 | CAN_L2 |
| | 6 | CAN_H2 |
| CAN1 | 7 | CAN_L1 |
| | 8 | CAN_H1 |

## 3.6 BUZZER（BUZZ）

- resources：PB20
- Low voltage effective(Active low)

## 3.7 USB OTG（J11）

- Mini USB connector
- resources（HDMB,HDPB）
- This interface connectors to the bottom board J11 and J19 connectors
- program can be written in or deleted from system via the miniUSB connector on core-board
- This port with OTG function, can be used to connect the USB device on the MINIPCI backplane (such as 3G module)

## 3.8 USB HOST（CN1）

- Standard Type A connector
- resources（HDMA、HDPA）
- Onboard Item NO：CN1
- High-speed USB 2.0 communication interface

## 3.9 TWI（J15）

- System provide a TWI interface
- Resources：PB10（TWD1），PB11（TWCK1）
- Interfaces：J15
- Pins definition：

| | | |
|---|---|---|
| TWI | 1 | TWD |
| | 2 | TWCK |
| | 3 | GND |
| | 4 | WKUP |

## 3.10 AUDIO OUT（J9）

- Standard 3.5mm audio in/out connector
- 32 Ω Output Load (HP_OUT)
- Resources：

| | | |
|---|---|---|
| AC97 (J9) | PD6 | AC97RX |
| | PD7 | AC97TK |
| | PD8 | AC97FS |
| | PD9 | AC97CK |

## 3.11 JTAG（J10）

| | | | |
|---|---|---|---|
| VDD33 | 1 | 2 | VDD33 |
| NTRST | 3 | 4 | GND |
| TDI | 5 | 6 | GND |
| TMS | 7 | 8 | GND |
| TCK | 9 | 10 | GND |
| RTCK | 11 | 12 | GND |
| TDO | 13 | 14 | GND |
| NRST | 15 | 16 | GND |
| | 17 | 18 | GND |
| | 19 | 20 | GND |

## 3.12 LCD（J20）

- 40PIN FPC connects to connector interface
- support interface mode 24/16bit mode (Default 16bit mode)
- Supports maximum definition of 1280*860
- This stand is located in the core board, Item NO:J5

CPU signal resources Note：

| | | | | |
|---|---|---|---|---|
| LCD | PE0 | LCDPWR | PE16 | LCDD9 |
| | PE1 | LCDMOD | PE17 | LCDD10 |
| | PE2 | LCDCC | PE18 | LCDD11 |
| | PE3 | LCDVSYNC | PE19 | LCDD12 |
| | PE4 | LCDHSYNC | PE20 | LCDD13 |

| | | | | |
|---|---|---|---|---|
| | PE5 | LCDDOTCK | PE21 | LCDD14 |
| | PE6 | LCDDEN | PE22 | LCDD15 |
| | PE7 | LCDD0 | PE23 | LCDD16 |
| | PE8 | LCDD1 | PE24 | LCDD17 |
| | PE9 | LCDD2 | PE25 | LCDD18 |
| | PE10 | LCDD3 | PE26 | LCDD19 |
| | PE11 | LCDD4 | PE27 | LCDD20 |
| | PE12 | LCDD5 | PE28 | LCDD21 |
| | PE13 | LCDD6 | PE29 | LCDD22 |
| | PE14 | LCDD7 | PE30 | LCDD23 |
| | PE15 | LCDD8 | | |

The connector of LCD interface located on the core-board, regarding to J5 on core-board, applies the upper connector, pin pitch of 0.5mm, and also integrates touch screen function. pin definition as below：

| | | | |
|---|---|---|---|
| VDD5V | 1 | 2 | VDD5V |
| B0 | 3 | 4 | B1 |
| B2 | 5 | 6 | B3 |
| B4 | 7 | 8 | B5 |
| B6 | 9 | 10 | B7 |
| GND | 11 | 12 | G0 |
| G1 | 13 | 14 | G2 |
| G3 | 15 | 16 | G4 |
| G5 | 17 | 18 | G6 |
| G7 | 19 | 20 | GND |
| R0 | 21 | 22 | R1 |
| R2 | 23 | 24 | R3 |
| R4 | 25 | 26 | R5 |
| R6 | 27 | 28 | R7 |
| GND | 29 | 30 | PWREN |
| LCDMOD | 31 | 32 | LCDCC |
| LCDDEN | 33 | 34 | VSYNC |
| HSYNC | 35 | 36 | DCLK |
| X- | 37 | 38 | X+ |
| Y- | 39 | 40 | Y+ |

LCD interface signal functional definition

## 3.13 KEY（B1）

- 2 keys：IO、TESET
- Item NO：B1:PB31 ,B4:RESET

## 3.14 TOUCH（J20）

- Touch interface comes from 4pin FPC connector
- Signal name：X+、X-、Y+、Y-
- resource：It comes from CPU
- functional resources：

| | | |
|---|---|---|
| TOUCH | PD20 | X+ |
| | PD21 | X- |
| | PD22 | Y+ |
| | PD23 | Y- |

The definition of interface J7 on Backplane pins are as below (the serial order of the two middle wire can be changed by wire jumper)：

| Pin | Function |
|---|---|
| 1 | Y- |
| 2 | X- (Y+) |
| 3 | Y+ (X-) |
| 4 | X+ |

## 3.15 LVDS（J3）

- Connector:20PIN（10*2）pin headers,2.0mm
- This interface via LCD interface TTL level transferred by the chip
- Support 24-bit mode
- This group of signals is converted into 24-bit LVDS interface signals by the chip of SN75LVDS83, which is located on the core board.the definition of J3 ports linked to back board by connector between core board and the backplane as below：

| | | | |
|---|---|---|---|
| VDD33V | 1 | 2 | VDD33V |
| GND | 3 | 4 | GND |
| TA- | 5 | 6 | TA+ |
| GND | 7 | 8 | TB- |
| TB+ | 9 | 10 | GND |
| TC- | 11 | 12 | TC+ |
| GND | 13 | 14 | TCLK- |
| TCLK+ | 15 | 16 | TD- |

| | | | |
|---|---|---|---|
| TD+ | 17 | 18 | |
| | 19 | 20 | VDD33V |

## 3.16 LED（D20）

- resources：PB21
- Low voltage effective(Active low)
- Wire connector J29 leads out the way IO , which is used as a system status indicator to use
- J29 pin is defined as follow：

| | | |
|---|---|---|
| LED | 1 | VDD33V |
| | 2 | PB22（470R） |
| | 3 | VDD33V |
| | 4 | GND（470R） |

## 3.17 CAN（J26）

- 2 CAN communication interface are extended through two groups of SPI controller
- CAN2.0 communication interface
- Resources ：

| | | | | | |
|---|---|---|---|---|---|
| CAN1 | PB0 | SPI0_MISO | CAN2 | PB14 | SPI1_MISO |
| | PB1 | SPI0_MOSI | | PB15 | SPI1_MOSI |
| | PB2 | SPI0_SPCK | | PB16 | SPI1_SPCK |
| | PD24 | SPI0_NPCS1 | | PD18 | SPI1_NPCS2 |
| | PD25 | INT | | PD26 | INT |

- CAN communication interface output to J27 Wiring connector is defined as follows：

| | | |
|---|---|---|
| COM3 | 1 | RRXD3 |
| | 2 | RTXD3 |
| | 3 | GND |
| CAN2 | 4 | GND_ISO |
| | 5 | CAN_L2 |
| | 6 | CAN_H2 |
| CAN1 | 7 | CAN_L1 |
| | 8 | CAN_H1 |

- If the establishment of CAN networking needs terminal resistance ,it can be achieved by applying the jumper cap J13, J14 of matching resistor resistor.
- (Note: This terminal matching resistance can only be used at both ends of UTP, meanwhile mid device does not require)

## 3.18 SPI（J31）

System provides a way SPI communication interface. Item NO：J31 pin are defined as follows:：

| | | |
|---|---|---|
| SPI | 1 | PB16 |
| | 2 | PB14 |
| | 3 | PB15 |
| | 4 | PD19 |
| | 5 | GND |

## 3.19 POWER OUT (J28)

- system provides a power output interface ,which can be used to provide power for the external LCD
- Specific output signal J28 following table：

| | | | |
|---|---|---|---|
| J28 | 1 | VDD12V | |
| | 2 | GND | |
| | 3 | ON_OFF | PB28 |
| | 4 | DIMMING | PD31 |
| | 5 | VDD5V | |

## 3.20 RTC

The system extends a high-definition RTC through TWI interface, the battery on the backplane can be charged automatically .extended resource as below：PA20（SDA），PA21（SCL）

# Chapter 4 System Electrical Description

- Power input：range（12-36V）
- Core board power supply：5V
- Kinds of power：+1.2V，+1.8V，+3.3V，+5V，+12V，
- Total power consumption：+12Vx0.25A (Without Screen)
- Actual test power consumption：
- Ambient temperature：-10-70℃

Core board connect to pin interface on bottom board is defined as follows:

| Pin(J8) | Signal | Pin(J8) | Signal | Pin(J6) | Signal | Pin(J6) | Signal |
|---|---|---|---|---|---|---|---|
| 1 | GND | 41 | GND | 1 | GND | 41 | SHDN |
| 2 | TA- | 42 | TA+ | 2 | 1V8 | 42 | PC1 |
| 3 | TB- | 43 | TB+ | 3 | PC0 | 43 | PC6 |
| 4 | TC- | 44 | TC+ | 4 | PC7 | 44 | PC9 |
| 5 | TD- | 45 | TD+ | 5 | PC10 | 45 | PC11 |
| 6 | TCLK- | 46 | TCLK+ | 6 | PC12 | 46 | PC13 |
| 7 | GND | 47 | VDD33 | 7 | PC15 | 47 | PC16 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 8 | PD24 | 48 | PD25 | 8 | PC17 | 48 | PC18 |
| 9 | PD26 | 49 | PD27 | 9 | PC19 | 49 | PC20 |
| 10 | PA6 | 50 | PE31 | 10 | DRRXD | 50 | DRTXD |
| 11 | PA8 | 51 | PA7 | 11 | PC21 | 51 | PC22 |
| 12 | PA20 | 52 | PA9 | 12 | PC23 | 52 | PC24 |
| 13 | PA22 | 53 | PA21 | 13 | PC25 | 53 | PC26 |
| 14 | PA24 | 54 | PA23 | 14 | PC27 | 54 | PC28 |
| 15 | PD20 | 55 | PD22 | 15 | PC29 | 55 | PC30 |
| 16 | PD21 | 56 | PD23 | 16 | PC31 | 56 | PD0 |
| 17 | PA28 | 57 | PA27 | 17 | PD6 | 57 | PD7 |
| 18 | PA30 | 58 | PA29 | 18 | PD8 | 58 | PD9 |
| 19 | PB0 | 59 | PA31 | 19 | GND | 59 | PD11 |
| 20 | PB2 | 60 | PB1 | 20 | GND | 60 | WKUP |
| 21 | PB5 | 61 | PB4 | 21 | PD12 | 61 | PD13 |
| 22 | PB7 | 62 | PB6 | 22 | PD14 | 62 | PD15 |
| 23 | PB9 | 63 | PB8 | 23 | PD16 | 63 | PD17 |
| 24 | PB11 | 64 | PB10 | 24 | PD18 | 64 | PD19 |
| 25 | PA26 | 65 | PA25 | 25 | PD29 | 65 | PD28 |
| 26 | | 66 | GND | 26 | PD31 | 66 | PD30 |
| 27 | PB15 | 67 | PB14 | 27 | PB20 | 67 | PB21 |
| 28 | PB17 | 68 | PB16 | 28 | PB22 | 68 | PB23 |
| 29 | PB19 | 69 | PB18 | 29 | PB24 | 69 | PB25 |
| 30 | NRST | 70 | VDD5V | 30 | PB26 | 70 | PB27 |
| 31 | PD10 | 71 | VSBAT | 31 | PB28 | 71 | PB29 |
| 32 | PA5 | 72 | PA4 | 32 | PB30 | 72 | NTRST |
| 33 | PA2 | 73 | PA1 | 33 | PB31 | 73 | TDI |
| 34 | PA3 | 74 | PA0 | 34 | RTCK | 74 | TDO |
| 35 | GND | 75 | PB3 | 35 | TCK | 75 | TMS |
| 36 | LED4 | 76 | LED5 | 36 | USB_B_VB | 76 | USB_A_VB |
| 37 | RX2- | 77 | RX2+ | 37 | HDPA | 77 | HDMA |
| 38 | VDD33 | 78 | AVDDT | 38 | VDD5V | 78 | |
| 39 | TX2- | 79 | TX2+ | 39 | HDPB | 79 | HDMB |
| 40 | GND | 80 | VDD5V | 40 | GND | 80 | GND |

Note：

In addition the functions of on the core board，but also through 160PIN double straight leads out functions as follows：

8-bit single channel LVDS interface (24-bit color)

Network port differential line

DEBUG serial port

USART0-3

USB OTG（HS）

USB 2.0
SPI*2
TWI*2
SDIO(HS)
PWM
ISI
AC97
TOUCH
JTAG
GPIOs

On core board there are two 2.54mm straight dual in line interface the pin signal define as:

U8 define as：(this interface can be used to connect large size LVDS LCD)

| | | | |
|---|---|---|---|
| TA- | 1 | 2 | TA+ |
| TB- | 3 | 4 | TB+ |
| TC- | 5 | 6 | TC+ |
| TD- | 7 | 8 | TD+ |
| TCLK- | 9 | 10 | TCLK+ |
| GND | 11 | 12 | VDD33 |
| DRTXD | 13 | 14 | VDD5V |
| DRRXD | 15 | 16 | GND |

U9 define as:：（This interface can be used as Ethernet interface）

| | | | |
|---|---|---|---|
| TX2- | 1 | 2 | TX2+ |
| RX2- | 3 | 4 | RX2+ |
| VDD33 | 5 | 6 | AVDDT |
| LED4 | 7 | 8 | LED5 |
| | 9 | 10 | |

Jumper on cpu core board：

J10 Dataflash CS.

J1 BMS pin of AT91SAM9G45 CPU.

J4 Nandflash CS

# Part 3 Linux System

## Chapter 5 Overview of Linux system

SBC6045 BSP mainly used for customization Linux for SBC6045 hardware platform, you can do secondary development based on the development kit. The SBC6045 CD provides the BSP packet that refers to table 1-1 content.

Table 1-1 specification of SBC6045 BSP

| Name | | Remark |
|---|---|---|
| BIOS | Bootstrap | NANDFLASH |
| | u-boot | NAND |
| | | NET and SAM-BA downloader image |
| Kernel | Linux-2.6.x | ROM/CRAM/EXT2/EXT3/FAT/NFS/ JFFS2/YAFFS2..etc |
| Device Driver | Serial | CPU DEBUG and four-way Serial |
| | RTC | AT91SAM9G45 extended RTC driver |
| | NET | 10/100M Ethernet driver |
| | Flash | Nand Flash driver |
| | LCD | TFT LCD driver for 480x272 800x480 and 800x600 Pixel LCD |
| | Touch Screen | The CPU touch screen control |
| | USB Host | USB Host driver |
| | USB Device | USB Device driver |
| | Watchdog | Watchdog driver |
| | SD Card | SD card driver |
| | EEPROM | 2K bits EEPROM |
| | CAN | Two-way extended CAN BUS driver |
| | RS-485 | One-way extended RS-485 |
| | LED | LED driver |
| | BEEP | Beep driver |
| Root file system | YAFFS2 | RW file system |
| Graphics | QT | Version number 4.4.2 |

# Chapter 6 Getting Start with SBC6045

*Note:If you use the HyperTerminal on the PC when SBC6045 is booting, please configure your connection software like this:*
*Baud rate:115200*
*Data: 8bit;*
*Parity:None*
*Stop: 1bit*
*Flow control: None*

## 6.1 Getting Start from Nandflash

The development board support booting from nandflash.If you want to boot from the nandflash, you should close the J4 on the board. When out factory, we have downloader image to the nandflash. If you want to update the image, you can refer to the "Chapter 9 Update the System Image".

*Note: The system image update can refer to【Chapter 9 Update the System Image】*

## 6.2 Setting the type of LCD Screen

Now the development board driver support three types of LCD screen,there are 480x272, 800x480, 800x600.
Set the tft parameter for the LCD.and the kernel can loader driver for the LCD.The tft param is follow:

| LCD screen size | LCD set parameter | LCD pixel |
|---|---|---|
| 4.3 inch | s | 480x272 |
| 7.0 inch | m | 800x480 |
| 10.4 inch | b | 800x600 |

Setting method is in the U-BOOT follow:

```
U-Boot> setenv bootargs console=ttySAC0,115200 tft=m root=/dev/mtdblock7 rw rootfstype=yaffs2
```

# Chapter 7 Test Tutorials of the SBC6045

## 7.1 Touch Screen Test

All about touch screen hardware connects can refer to "Hardware manual of SBC6045 "
1. Enter the following command procedures for the implementation of touch screen calibration:

```
[root@Mini6045:/]# ts_calibrate
```

Follow the on-screen prompts, click the "+" icon five times to complete the calibration.
2. Upon completion of calibration, enter the following commands for touch-screen test:

```
[root@Mini6045:/]# ts_test
```

Follow the on-screen prompts, can choice of "draw point' or "draw line" test.

*Note: pressing the Ctrl+C then exit the test program.*

## 7.2 Net Test

There have a MACB 10/100M Ethernet on the development board, users can use it connect to network, and use following command to test:

```
~ $   ifconfig eth0 192.192.192.200
~ $   ping 192.192.192.105
PING 192.192.192.105 (192.192.192.105): 56 data bytes
64 bytes from 192.192.192.105: icmp_seq=0 ttl=64 time=0.5 ms
64 bytes from 192.192.192.105: icmp_seq=1 ttl=64 time=0.3 ms
64 bytes from 192.192.192.105: icmp_seq=2 ttl=64 time=0.3 ms
64 bytes from 192.192.192.105: icmp_seq=3 ttl=64 time=0.3 ms
64 bytes from 192.192.192.105: icmp_seq=4 ttl=64 time=0.3 ms
64 bytes from 192.192.192.105: icmp_seq=5 ttl=64 time=0.3 ms
64 bytes from 192.192.192.105: icmp_seq=6 ttl=64 time=0.3 ms

--- 192.192.192.105 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.3/0.5 ms
~ $
```

*Note: pressing the Ctrl+C then exit the test program.*

## 7.3 UART Test

There are five serial ports on the development board,which CPU has 4,and each device namded for linux is “ttySAC0,ttySAC1,ttySAC2, ttySAC3 and ttySAC4” . “ttySAC0” is used for debug, “ttySAC1” and “ttySAC3” is used for RS-485(They also could be used for Serial). You can do test with the following commands. All about serial ports connect can refer to “Hardware manual of SBC6045”

```
[root@Mini6045:/home/app]# ./com -d /dev/ttySAC1 -s 1234567890
```

## 7.4 CAN Bus Test

There are two extended CAN Bus device on the development board. All about the CAN Bus hardware connect can refer to “Hardware manual of SBC6045” .

1. CAN test, type follow command in hyper terminal window:

```
[root@Mini6045:/]# cd /home/app/can/
[root@Mini6045:/home/app/can]# . / can_test_125k /dev/can0
```

The result information as follow:

```
/dev/can0 Work Rate: 125000bps
/dev/can0 work on normal mode.

Press "\q" to quit!
Input the message and press Enter to send!
The frame id=1...2031, and eid=id*2+1

CanTest> TestData
CanTest>
```

```
Header: id=1 srr=0 ide=0 eid=0 rtr=0 rb1=0 rb0=0 dlc=8
RECV: "TestData"
```

2. Loop back mode test.

```
[root@Mini6045:/home/app/can]# ./can_test_125k /dev/can0   -f
```

*Notes: Press Ctrl+C to terminate.*

## 7.5 RS-485 Test

The RS-485 takes up serial device /dev/ttySAC1 and /dev/ttySAC3. Test methods, such as serial port can be tested solution, but only in hardware connect is different.

## 7.6 USB Host Test

There only one USB host device on the development board. Test the USB host will need a USB device (For example, a USB disk). When you plug in a USB device, system will print related information on your Hyper Terminal.

```
usb 1-1: USB disconnect, address 2
usb 1-1: new full speed USB device using at91_ohci and address 3
usb 1-1: configuration #1 chosen from 1 choice
scsi2 : SCSI emulation for USB Mass Storage devices
scsi 2:0:0:0: Direct-Access     Generic  USB  SD Reader   0.00 PQ: 0 ANSI: 2
sd 2:0:0:0: [sda] 7744512 512-byte hardware sectors (3965 MB)
sd 2:0:0:0: [sda] Write Protect is off
sd 2:0:0:0: [sda] Assuming drive cache: write through
sd 2:0:0:0: [sda] 7744512 512-byte hardware sectors (3965 MB)
sd 2:0:0:0: [sda] Write Protect is off
sd 2:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1
sd 2:0:0:0: [sdb] Attached SCSI removable disk
sd 2:0:0:0: Attached scsi generic sg1 type 0
```

Prompted by the above information can be seen that usb disk device was identified as sda1. You can do test with the following command.

1. Users can opera the usb disk after mount u-disk by following command, because Linux system is request must have a directory was used by usb disk.For example, you can mount it into /mnt directory and set vfat format.

```
[root@Mini6045:/]# mount -t vfat /dev/sda1 /mnt/
```

2. Enter into the directory and view files.

```
[root@Mini6045:/]# cd /mnt/
[root@Mini6045:/mnt]# ls
```

3.Umount the usb disk

```
[root@Mini6045:/mnt]# cd /
[root@Mini6045:/]# umount /mnt/
```

*Note: usb disk will identify as sda device by default, you can seen it from system prompt.*

## 7.7 RTC Test

The development board has hardware clock chip used for store/restore system time. You can opera the RTC like this:

1. View system clock information

```
[root@Mini6045:/]# date
Thu May 27 11:48:02 UTC 2010
```

2. Change the date, like May 27 11:49 2010.

```
[root@Mini6045:/]# date 052711492010
Thu May 27 11:49:00 UTC 2010
```

3.Save the system clock to RTC

```
[root@Mini6045:/]# hwclock -w
```

4. View RTC clock information

```
[root@Mini6045:/]# hwclock -r
Thu May 27 11:49:07 2010    0.000000 seconds
```

5. Restore the RTC backup to system.

```
[root@Mini6045:/]# hwclock -s
[root@Mini6045:/]# date
Thu May 27 11:49:45 UTC 2010
```

## 7.8 EEPROM Test

Under Linux system, the EEPROM device was regarded as a virtual file, you can use the following command to operate the EEPROM.

```
[root@Mini6045:/]# eeprom –help
Usage: eeprom read <addr> <size> --read size bytes at addr
       eeprom write <hex> <addr> --write 1 byte value to addr
```

【Attention】address and date automatically regard as hex date.

1 Write data to EEPROM . ex:write 0x05 to addr 0xfa.

```
[root@Mini6045:/]# eeprom write 0xfa 0x05
```

2 Read data from EEPROM . ex:Read one byte from0x05.

```
[root@Mini6045:/]# eeprom read 0x05 1
```

## 7.9 SD Card Test

Insert in SD card, the system will print the following information. User can execute the following mount command to mount SD card and access SD card.

```
[root@Mini6045:/]# mmc1: new SD card at address 0002
mmcblk0: mmc1:0002 N/A     489 MiB
```

```
mmcblk0: p1
```

Prompted by the above information can be seen that SD card was identified as mmcblk0p1. You can do test with the following command.

1. Users can opera the SD card after mount SD card by following command, because Linux system is request must have a directory was used by SD card . For example, you can mount it into /mnt directory and set vfat format.

```
[root@Mini6045:/]# mount -t vfat /dev/mmcblk0p1 /mnt/
```

2. Enter into the directory and view files.

```
[root@Mini6045:/]# cd /mnt/
[root@Mini6045:/mnt]# ls
```

3. Umount theSD card

```
[root@Mini6045:/mnt]# cd /
[root@Mini6045:/]# umount /mnt/
```

*Note: SD card will identify as mmcblk0p1 device by default, you can seen it from system prompt.*

## 7.10 LED Test

There are some leds on the development board, they are used as indication. You can do test with following command.

1. Turn on led (D20)

```
[root@Mini6045:/]# echo '0' >/sys/class/leds/d7/brightness
```

2. Turn off led (D20)

```
[root@Mini6045:/]# echo '1' >/sys/class/leds/d7/brightness
```

## 7.11 Beep Test

There have a Beep driving circuit on the development board. You can operate the Beep like this:

1. Beep ring

```
[root@Mini6045:/]# echo '1' >/sys/class/leds/beep/brightness
```

2. Beep stop

```
[root@Mini6045:/]# echo '0' >/sys/class/leds/beep/brightness
```

# Chapter 8 Setting the Development Environment

Before development the SBC6045, users must setup arm-linux cross development environment. Following the example of Ubuntu operating system(Linux).

## 8.1 Install the Cross Compiler

If you have not installed arm-none-linux-gnueabi-gcc, copy "/media/cdrom/linux/tools/arm-2007q1-10-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2" (from

CD) to the folder of Linux system.

The installation command is:

```
mkdir /usr/local/arm
tar –jxvf arm-2007q1-10-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2 –C /usr/local/arm
```

## 8.2 Setting the Cross-compiler Environment

Define the path of compiler after installation is completed.

```
export PATH=/usr/local/arm/arm-2007q1/bin/:$PATH
```

*Note:You can copy it to ".bashrc" file, that it'll add the PATH automatically when OS is booting.*

## 8.3 Compiler the System of the SBC6045

### 8.3.1 Ready to Compiler Files

All components of the system's source is located in "linux/source" directory, you must decompression the files before development.

For example:

```
root@LINUXSERVER:~# mkdir embest
root@LINUXSERVER:~# cd embest/
root@LINUXSERVER:~/embest# cp /media/cdrom/02\ Linux\ 2.6\ Kit/01\ SourceCode/bootloader/Bootstrap-v1.14.tar.bz2 ./
root@LINUXSERVER:~/embest# cp /media/cdrom/02\ Linux\ 2.6\ Kit/01\ SourceCode/bootloader/u-boot-1.3.4.tar.bz2 ./
root@LINUXSERVER:~/embest# cp /media/02\ Linux\ 2.6\ Kit/01\ SourceCode/kernel /linux-2.6.30.tar.bz2 ./
root@LINUXSERVER:~/embest#
    cp /media/cdrom/02\ Linux\ 2.6\ Kit/01\ SourceCode/rfs/rfs-qtopia.tar.bz2 ./
root@LINUXSERVER:~/embest# mkdir tools
root@LINUXSERVER:~/embest# cd tools/
root@LINUXSERVER:~/embest/tools#
cp /media/cdrom/02\ Linux\ 2.6\ Kit/02\ Tools/mkyaffs2image ./
root@LINUXSERVER:~/embest/tools#
cp /media/cdrom/02\ Linux\ 2.6\ Kit/02\ Tools/mkimage ./
root@LINUXSERVER:~/embest/tools# export PATH=/root/embest/tools:$PATH
root@LINUXSERVER:~/embest/tools# cd ../
root@LINUXSERVER:~/embest# tar jxvf Bootstrap-v1.14.tar.bz2
root@LINUXSERVER:~/embest# tar jxvf u-boot-1.3.4.tar.bz2
root@LINUXSERVER:~/embest# tar jxvf linux-2.6.30.tar.bz2
root@LINUXSERVER:~/embest# tar jxvf rfs-qtopia.tar.bz2
```

When operation for complete, that you can see linux-2.6.30、u-boot-1.3.4、Bootstrap-v1.14、rfs-qtopia four directory will be generated under the current folder.

### 8.3.2 The First Booting Stage -AT91Bootstrap

SBC6045 development board is support for boot from nandflash, following commands will be generated nandflash_at91sam9g45ekes.bin binary file.

```
root@LINUXSERVER:~/embest# cd Bootstrap-v1.14
root@LINUXSERVER:~/embest/Bootstrap-v1.14# cd board/at91sam9g45ekes/nandflash
root@LINUXSERVER:~/embest/Bootstrap-v1.14/board/at91sam9g45ekes/nandflash# make
```

When operation for complete,we need nandflash_at91sam9g45ekes.bin binary file in the current directory.

### 8.3.3 The Second Booting Stage-U-boot

U-boot is used in the second booting stage of SBC6045, enter the following commands in the terminal, u-boot.bin will be generated under current folder.

```
root@LINUXSERVER:~/embest/u-boot-1.3.4# make at91sam9g45ekes_nandflash_config
root@LINUXSERVER:~/embest/u-boot-1.3.4# make
```

When operation for complete, we need u-boot.bin binary file in the current directory.

### 8.3.4 The Third Booting Stage-Kernel

You can use the follow commands to complier kernel:

```
root@LINUXSERVER:~/embest/linux-2.6.30#
    cp arch/arm/configs/SBC6045 _defconfig .config
root@LINUXSERVER:~/embest/linux-2.6.30# make menuconfig
root@LINUXSERVER:~/embest/linux-2.6.30# make uImage
```

Wait the compilation of uImage to be finished, and found it under /arch/arm/boot/ folder.

### 8.3.5 Make Root File System Image

Use the follow commands to generated rfs.yaffs2 file as we need with tool program "*02 Linux 2.6 Kit\02 Tools\mkyaffs2image*" in CD.

```
root@LINUXSERVER:~/embest# mkyaffs2image rfs/ rfs.yaffs2
```

## 8.4 System Customization

In fact, there are many kernel configuration options at Linux kernel, users can Increase or reduction of kernel features, make better suited to the needs of users. The next chapter is an example for kernel configuration process.

Kernel source code in the factory provided the default configuration file:

arch/arm/configs/SBC6045_defconfig

Users can custom system base on the default configuration file.

```
root@LINUXSERVER:~/embest/linux-2.6.30#
    cp arch/arm/configs/SBC6045_defconfig .config
root@LINUXSERVER:~/embest/linux-2.6.30# make menuconfig
```

The following configure for each driver Help:

**RTC Driver:**

1.Select Device drivers

2.Select Real Time Clock

3.Select Maxim/Dallas DS3231

**Watchdog Driver**

1.Select Device drivers

2.Select Watchdog Timer Support

3.Select AT91SAM9 watchdog

**MACB Driver**

1.Select Device Drivers

2.Select Network device support

3.Select Ethernet(10 or 100Mbit)

3.Select Atmel MACB support

**Graphics support**

1.Select Device Drivers

2.Select Graphics support

3.Select Support for frame buffer devices

4.Select AT91/AT32 LCD Controller support

**CAN 驱动**

1.Select Device drivers

2. Select CAN support

3. Select CAN support

**Touch Screen Driver**

1.Select Device Drivers

2.Select Input device support

3.Select Touchscreens

4.Select Atmel Touchscreen Interface

Save the config,compile the kernel by follow command:

```
root@LINUXSERVER:~/embest/linux-2.6.24# make uImage
```

# Chapter 9 Update the System Image

The development board with Nandflash, currently supports nandflash from startup.

## 9.1 System Image Map

The Map of Nandflash:

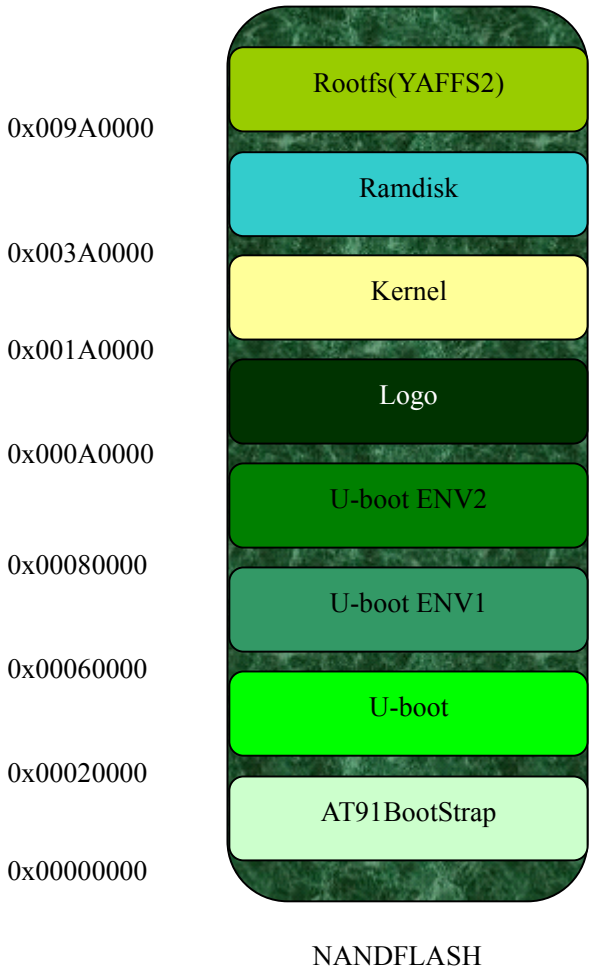



Figure 5.1 Nandflash image map

## 9.2 Setting the Environment of Burning Image

Install the application *02 Linux2.6 Kit\02 Tools\Install AT91-ISP v1.13.exe* (or higher) in your PC. You will see the icon *SAM-BA v2.9* appearing on the desktop of PC.

## 9.3 Burning System Image

### 9.3.1 Burn the Image of the first stage –AT91BootStarp

- Connect debug serial port. About hardware connects can refer to "Hardware manual of SBC6045 ".
- Connect USB to your PC using USB cable.
- Make sure no SD card in the development board.
- Open the HyperTerminalon the PC, config the baud 115200, 8 bit data, 1 bit stop, none parity, none Flow control.
- Before power on, open the **J1**, **J4** and **J10**. Then power on.
- Double click the ***SAMBA v2.9*** cursor on the desktop of the PC.


Figure 9.3.1 SAM-BA v2.9 software icons

➢ Shown in the figure below, select "***at91sam9g45-ek***"and "***\usb\ARM0***", and click "*Connect*".

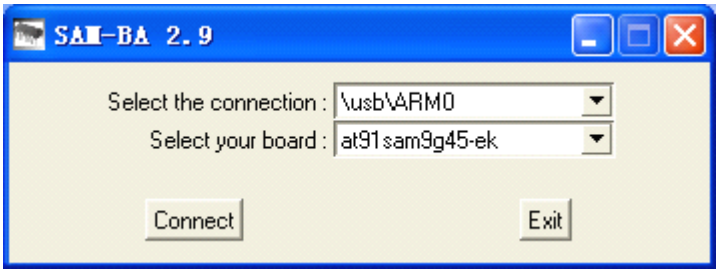



Figure 9.3.2 SAM-BA startup interface

➢ After click "***Connect***", then entering SAM-BA software.

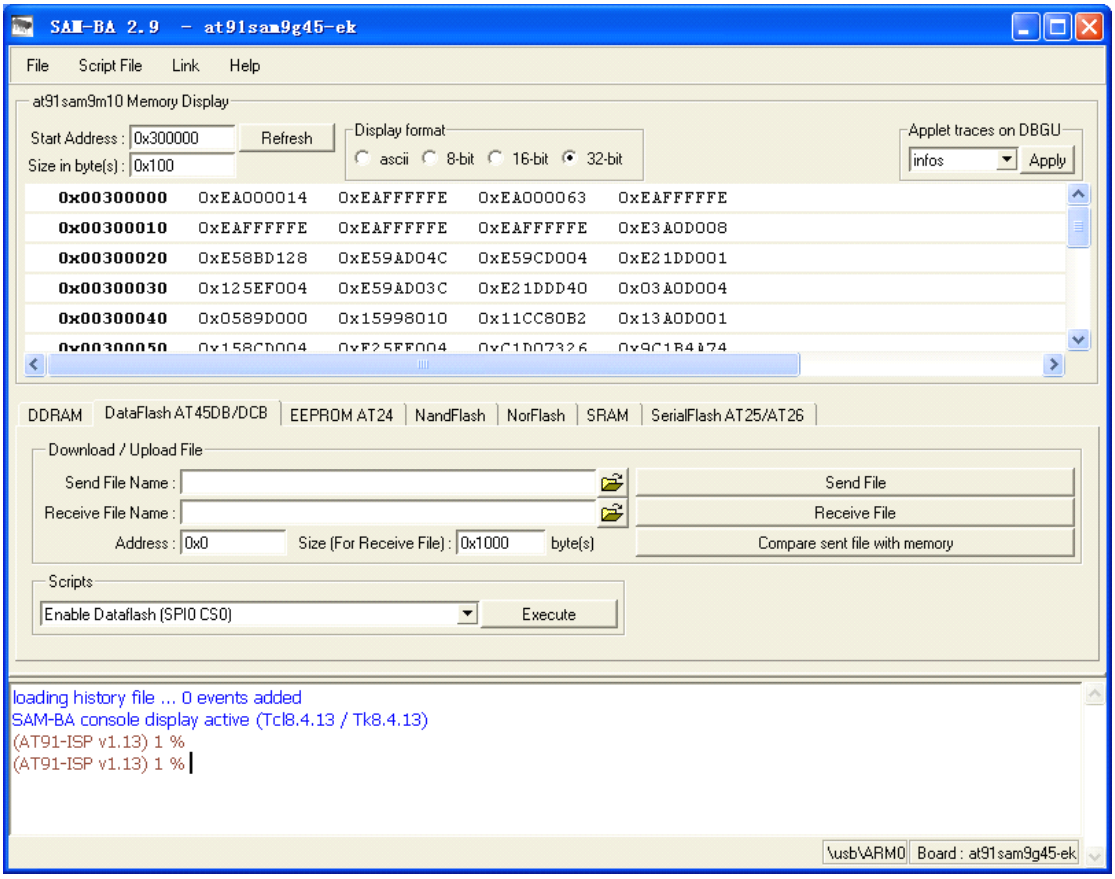



Figure 9.3.3 SAM-BA burn interface

➢ Close the **J4**, select "***nandflash***" tab. Then select the options "***Enable NandFlash***" at "***Scripts***" Pull-down menus, then click "***Execute***". Shown in the figure below:

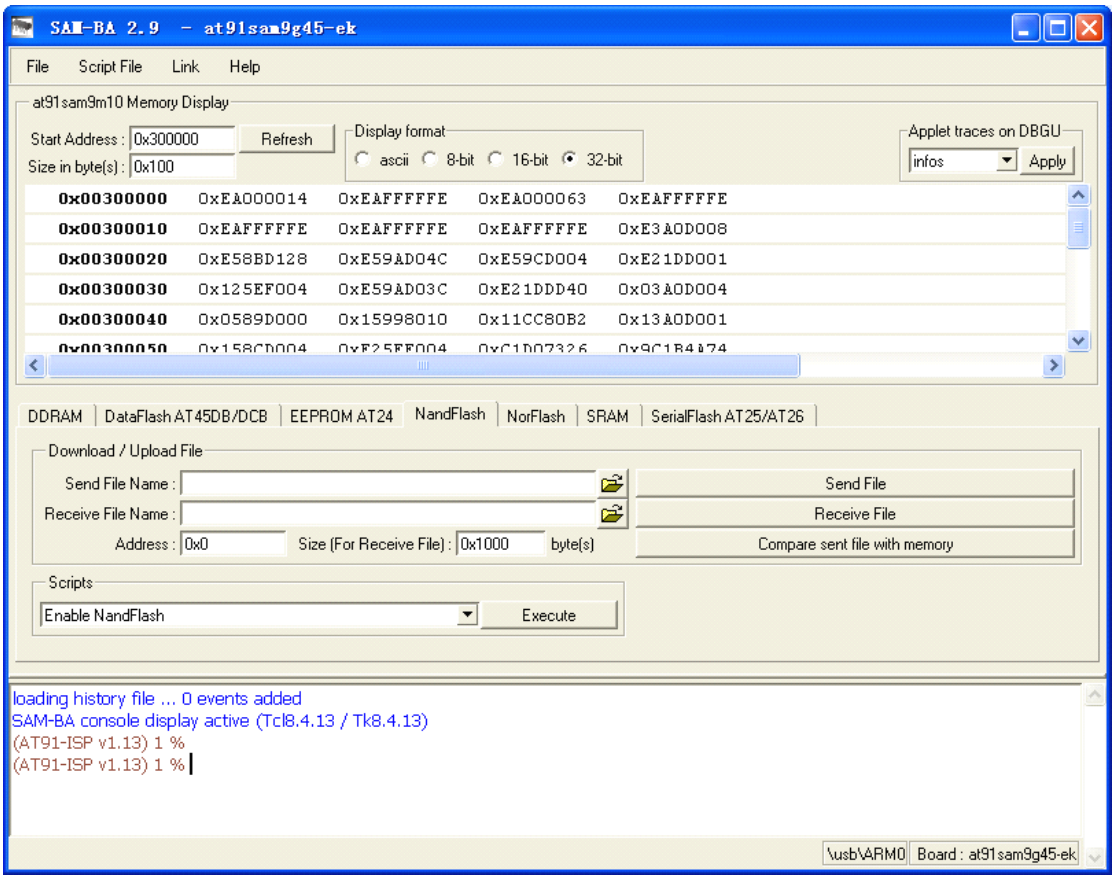



Figure 9.3.4 SAM-BA enable the nandflash

- At "***Scripts***" Pull-down menus, select "***Erase All***", and click "***Execute***", wait for a moment, you can see nandflash will be erased.

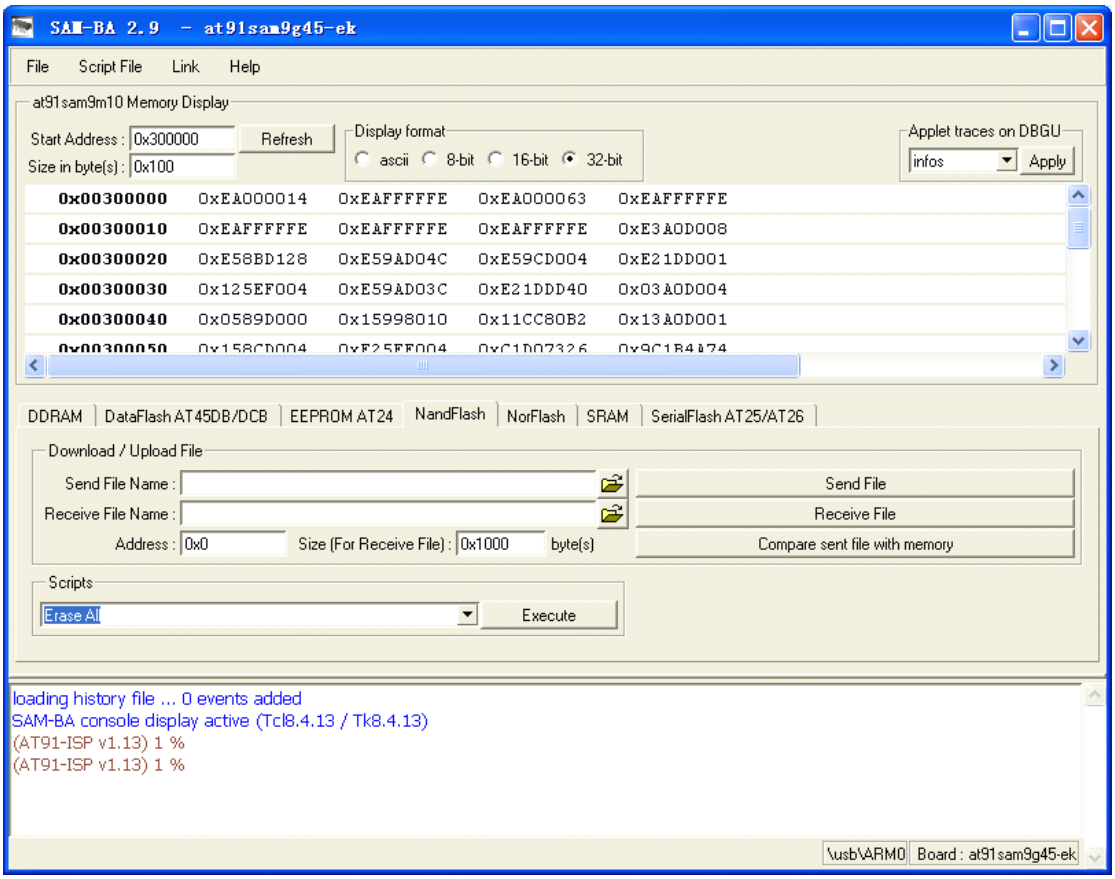



Figure 9.3.5 SAM-BA erase nandflash interface

- At "***Scripts***" Pull-down menus, select "***Scrub NandFlash***", and click "***Execute***", wait for a moment, you can see nandflash will be Erased.

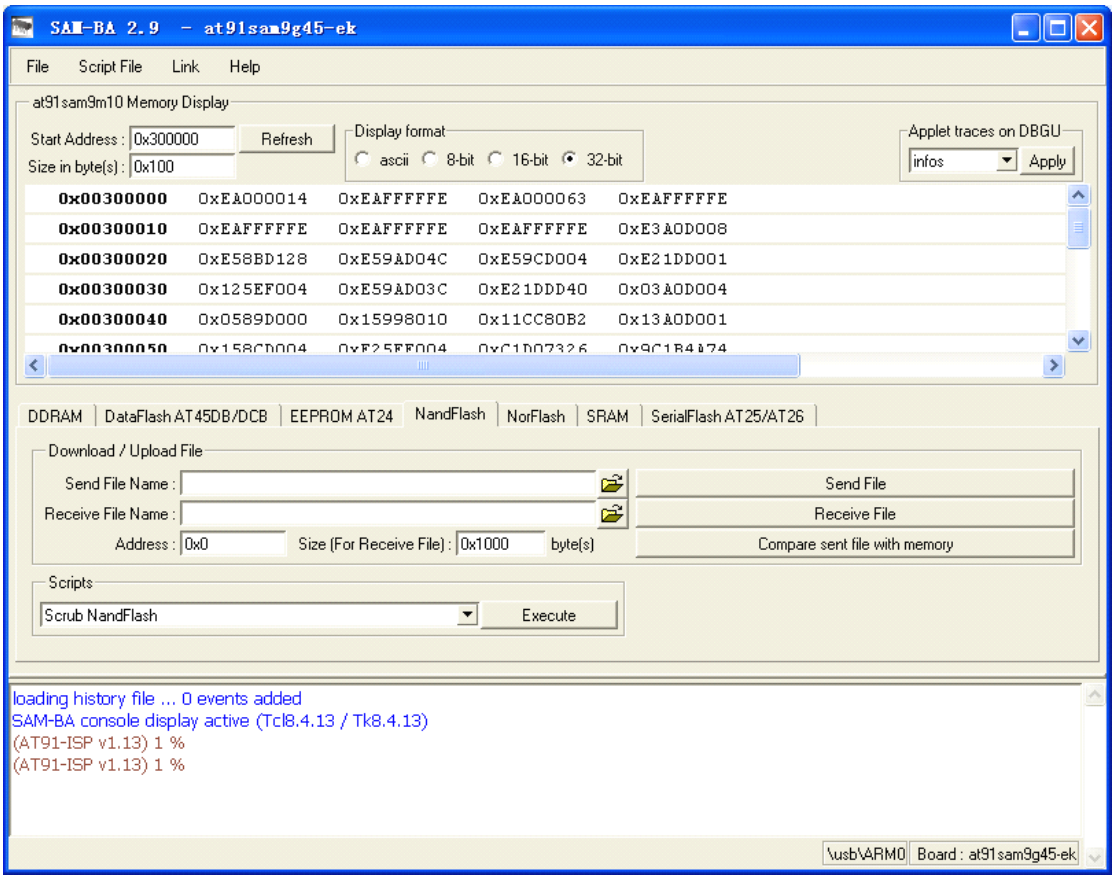



Figure 9.3.6 SAM-BA erase nandflash interface

➢ At "***Scripts***" Pull-down menus, select "***Send Boot File***", and click "***Execute***".

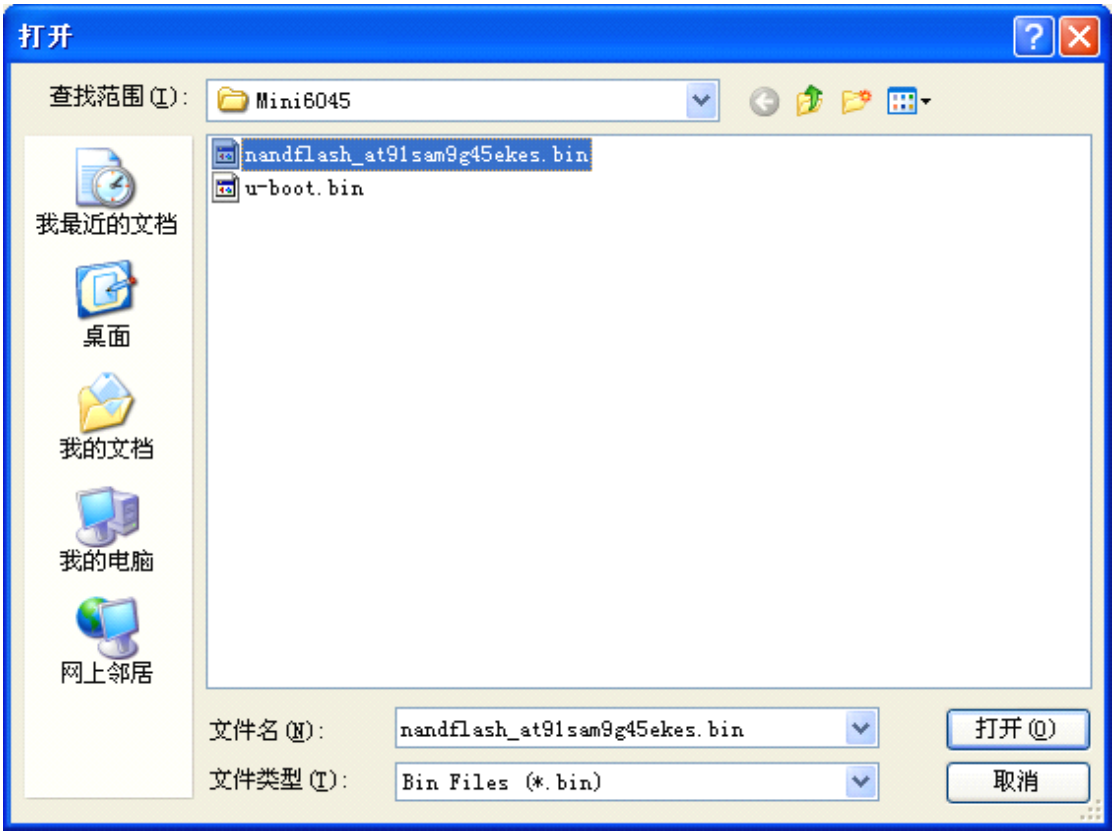



Figure 9.3.7 SAM-BA burn nandlash_at91sam9g45ekes.bin

➢ When you click "***Execute***", you will open a dialog box, select "***nandflash_at91sam9g45ekes.bin***" and click open. Then you can download this file to target board.

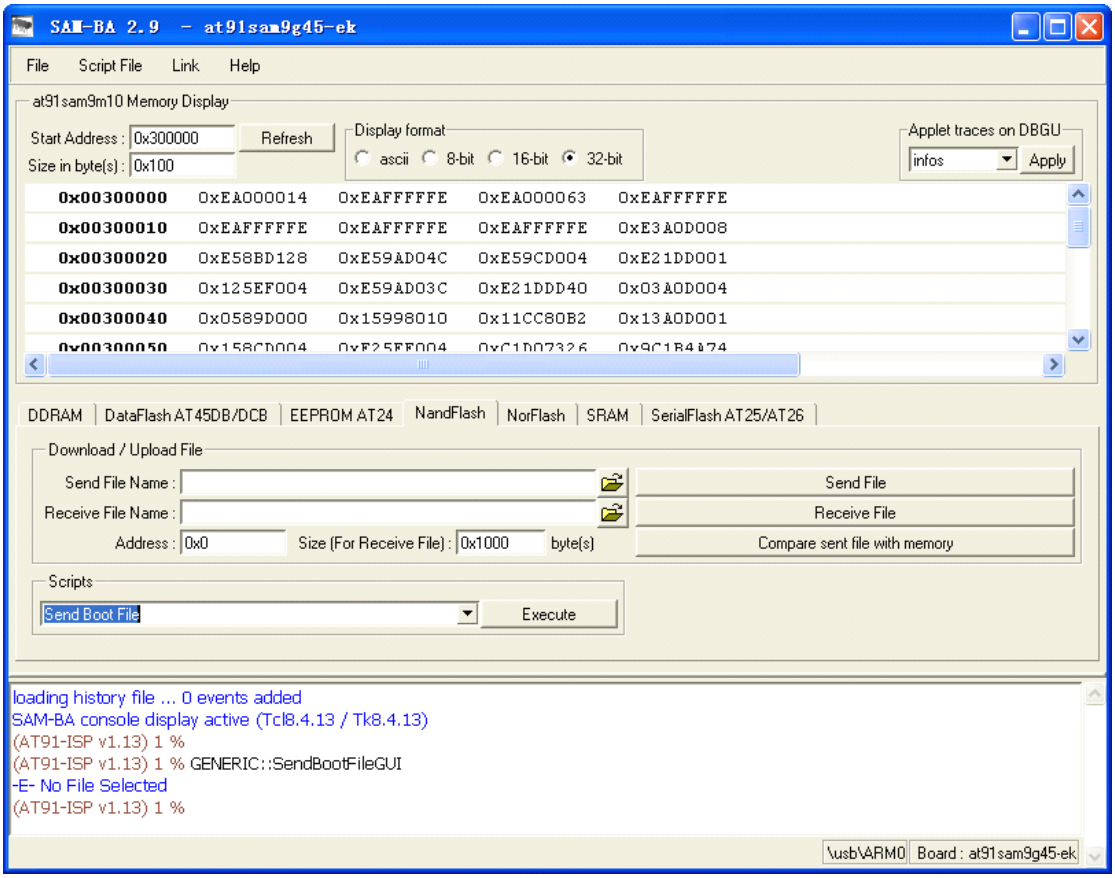



Figure 9.3.8 SAM-BA burn nand_at91sam9g45ekes.bin

## 9.3.2 Burn the Image of second booting stage u-boot.

Burn the u-boot.bin file to nandflash through SAM-BA.

➢ Enter the downloader address of U-boot in the "***address***" field: "***0x20000***"

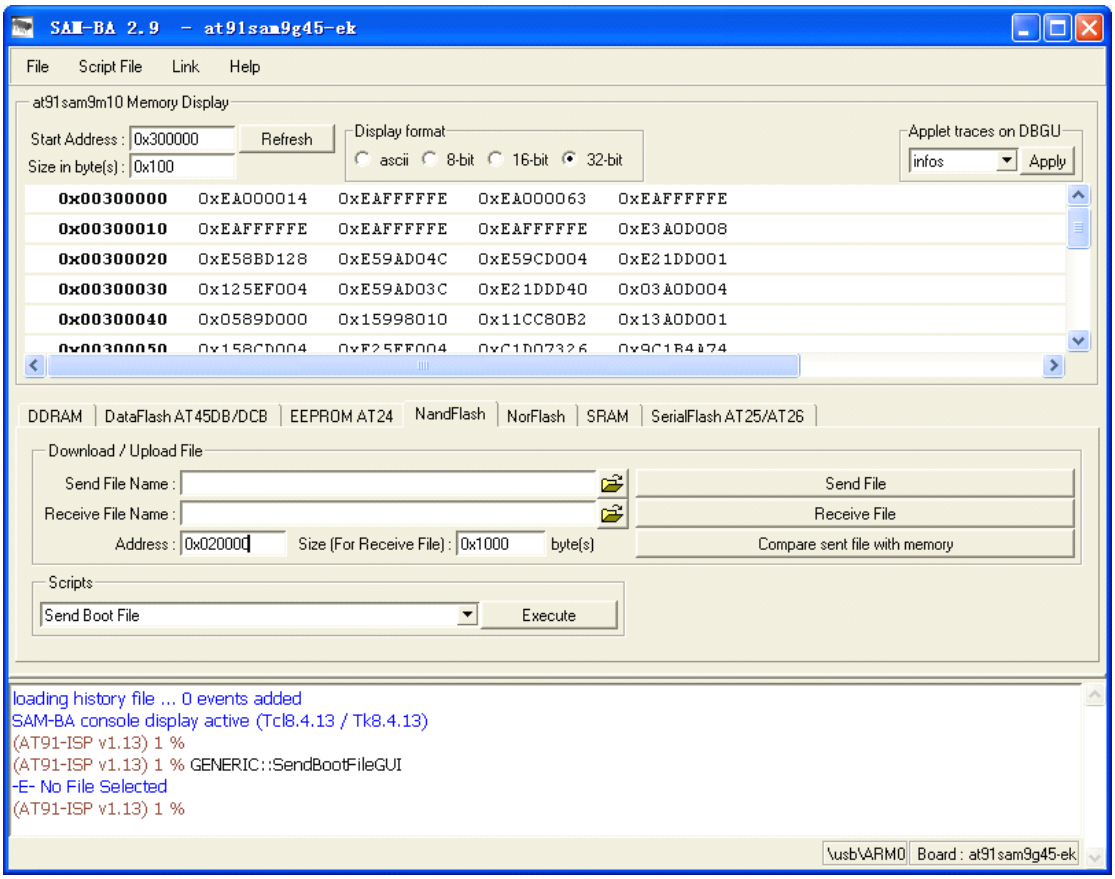



Figure 9.3.9 SAM-BA burn u-boot.bin

➢ Open the "***Send File Name***" browse window, click "***open folder***" button.

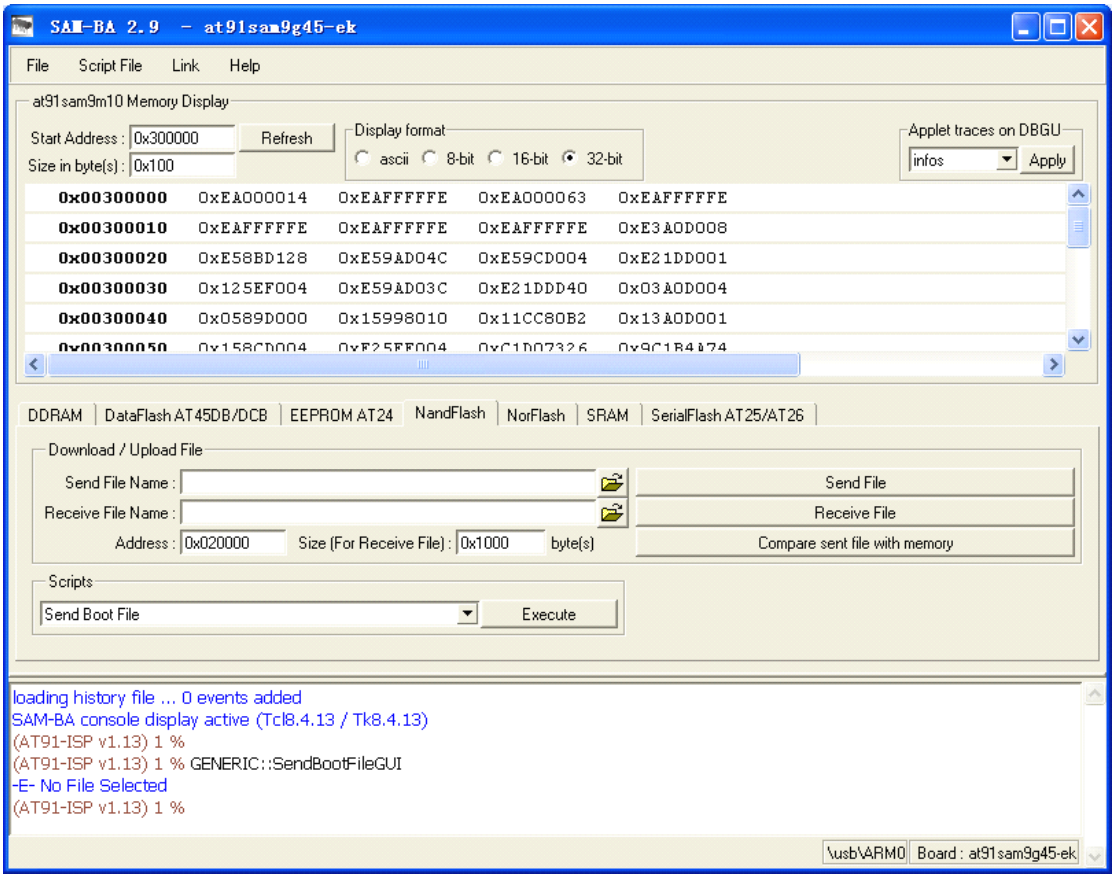


Figure 9.3.10 SAM-BA burn u-boot.bin

- You will open a dialog box, select "***u-boot.bin***" and click open

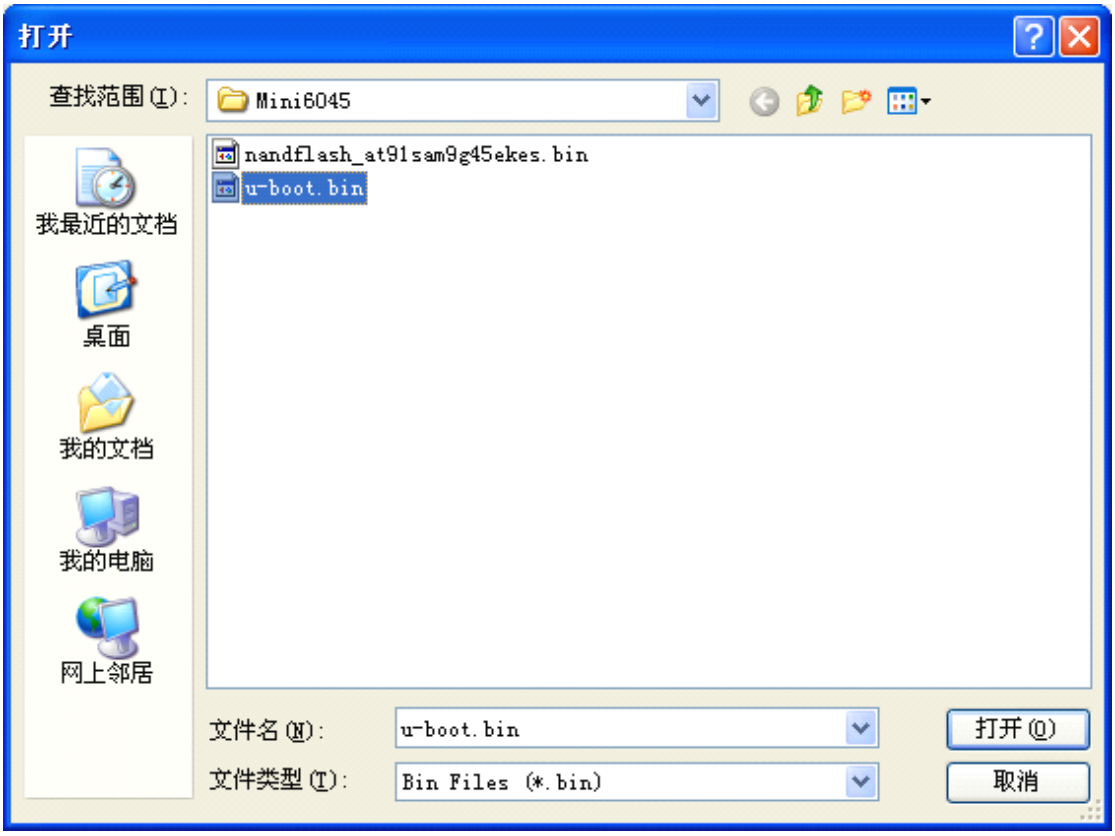



Figure 9.3.11 SAM-BA burn u-boot.bin

- Then download "***u-boot.bin***" to target board dataflash, click "***Send File***" button to burn "u-boot.bin" file to nandflash.

### 9.3.3 Burn the Logo Image –Logo

Burn "logo.bin" file to nandflash through SAM-BA.

- Enter the downloader address of uImage in the "***address***" field:"***0xA0000***".

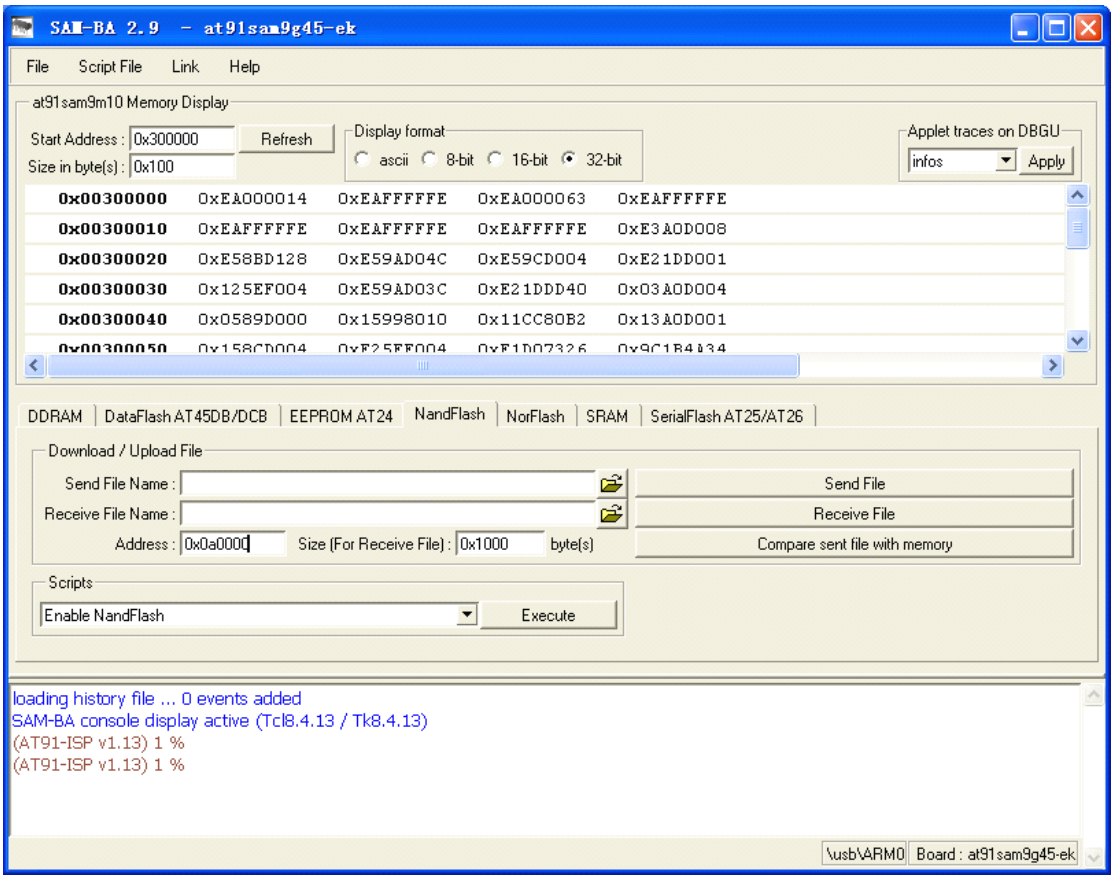



Figure 9.3.12 SAM-BA burn logo.bin to nandflash

➢ Open the "***Send File Name***" browse window, click "***open folder***" button.

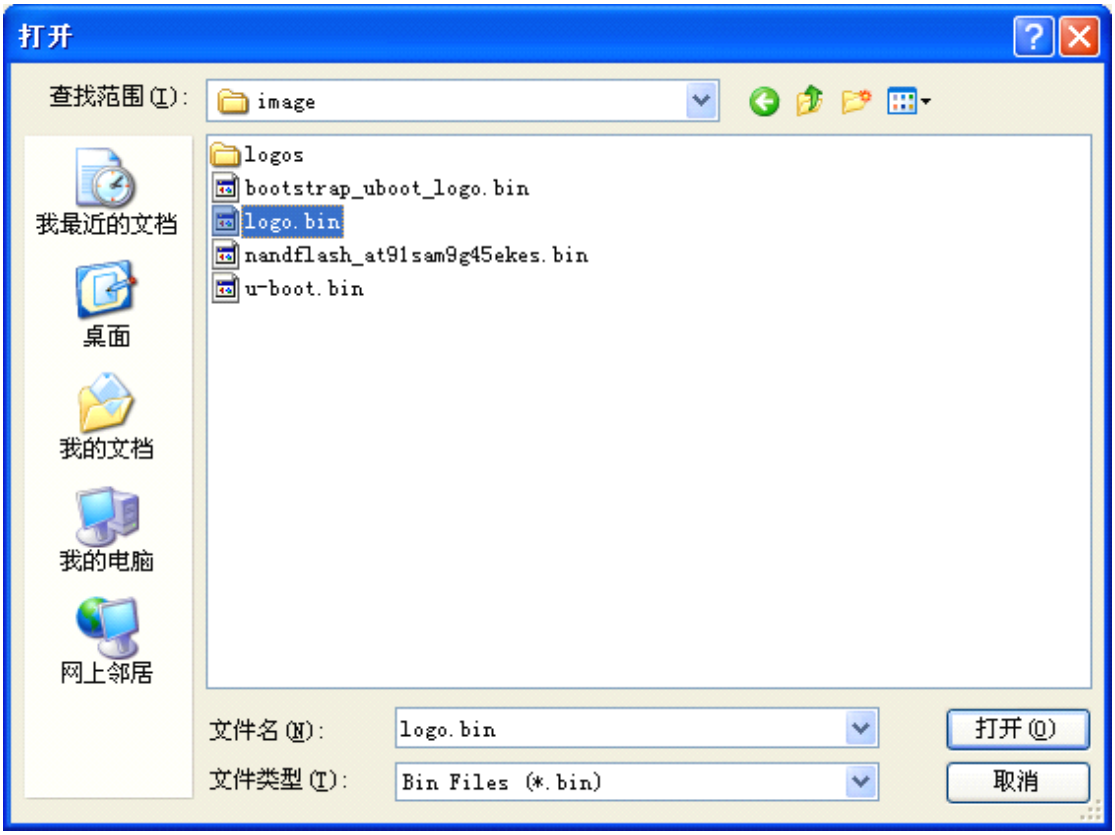



Figure 9.3.13 SAM-BA burn logo.bin to nandflash

- Open the "***Send File Name***" browse window, select "***logo.bin***",as shown in the figure below.
- Click "***Send File***" button to burn "***logo.bin***" to nandflash.

### 9.3.4 Burn the Image of the third booting –kernel

Burn "uImage" file to nandflash through SAM-BA.

- Enter the downloader address of uImage in the "***address***" field:"***0x1A0000***".

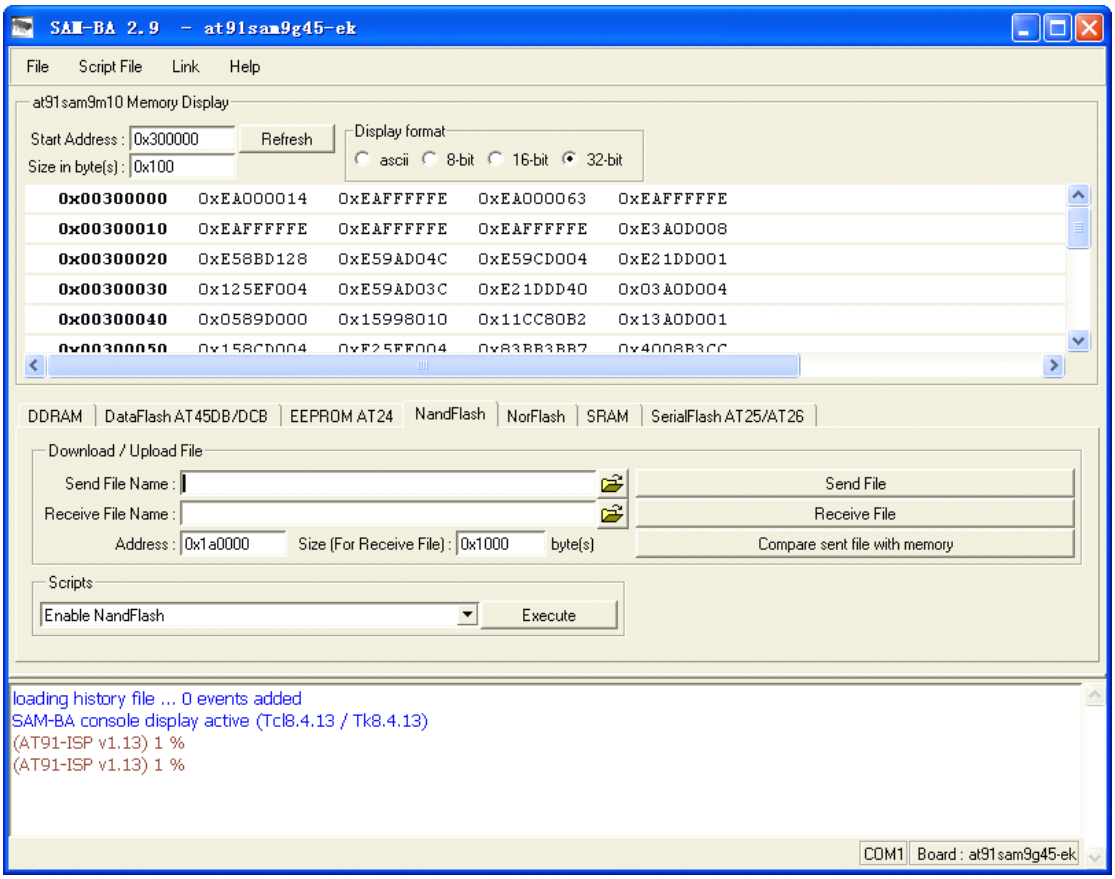



Figure 9.3.14 SAM-BA burn uImage to nandflash

- Open the "***Send File Name***" browse window, click "***open folder***" button.

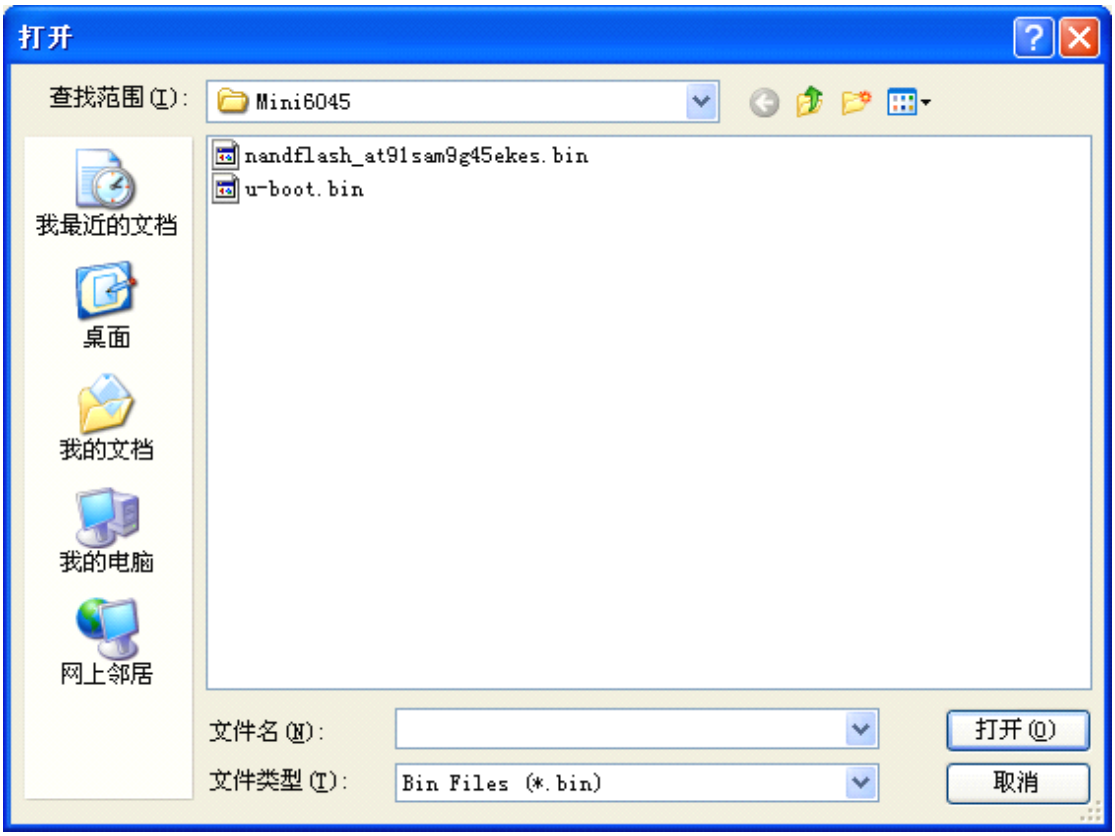



Figure 9.3.15 SAM-BA burn uImage binary file

➢ Open the "***Send File Name***" browse window, then you should select "***All Files (*.*)***" in the "***files of type***", select "***uImage***",as shown in the figure below.

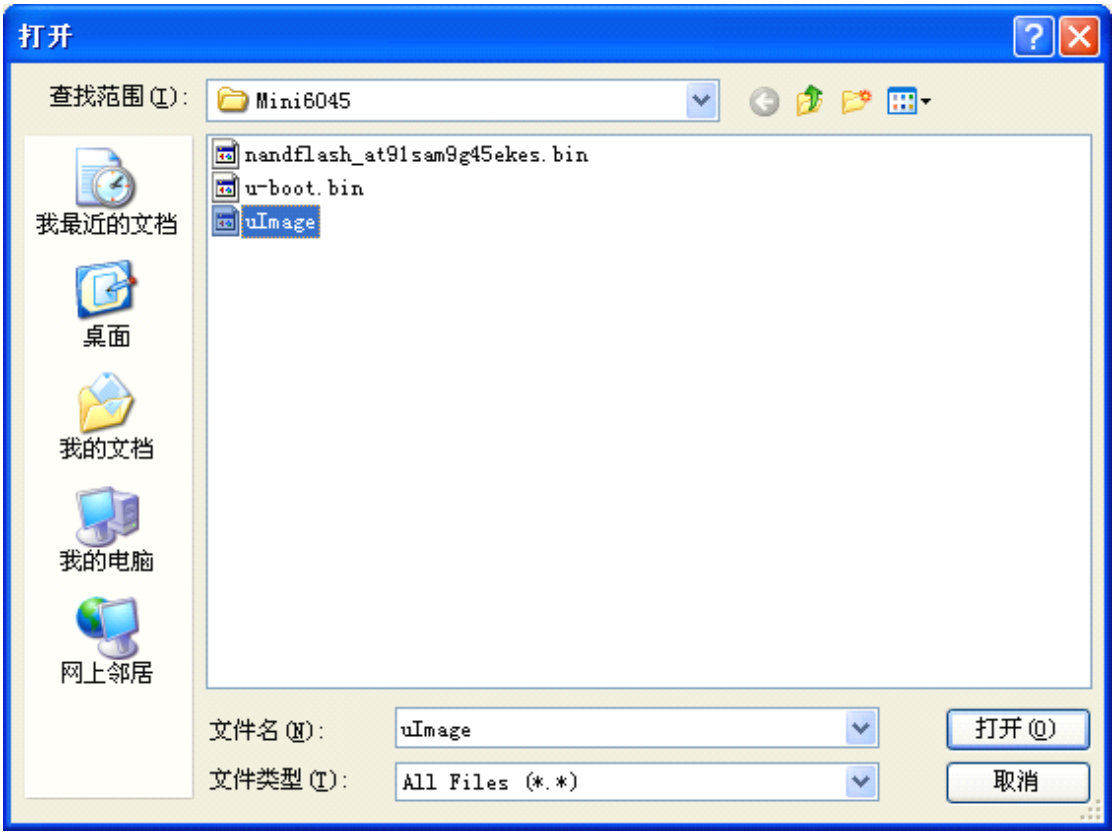



Figure 9.3.1.16 SAM-BA burn uImage to nandflash

- Click "***Send File***" button to burn "uImage" to nandflash.

### 9.3.5 Burn the Image of the fourth booting stage –file system

- Make target board connect to network.
- Set tftpd server and download folder. Copy tftpd.exe in the CD's ***02 Linux2.6 Kit/02 Tools*** folder to your PC.

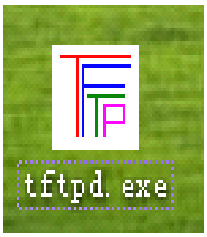



Figure 9.3.17 Tftpd.exe software icon

- Double click "*tftpd.exe* ", the following picture will appear on the screen.


Figure 9.3.18 Tftpd.exe open interface

- Click "*Tftpd* ", and click the option "*Configure* " in its pulldown menu.


Figure 9.3.19 Tftpd.exe config interface

- Press the button *Browse*, the following picture will appear on the screen, select "image record", and click "*OK* " button.

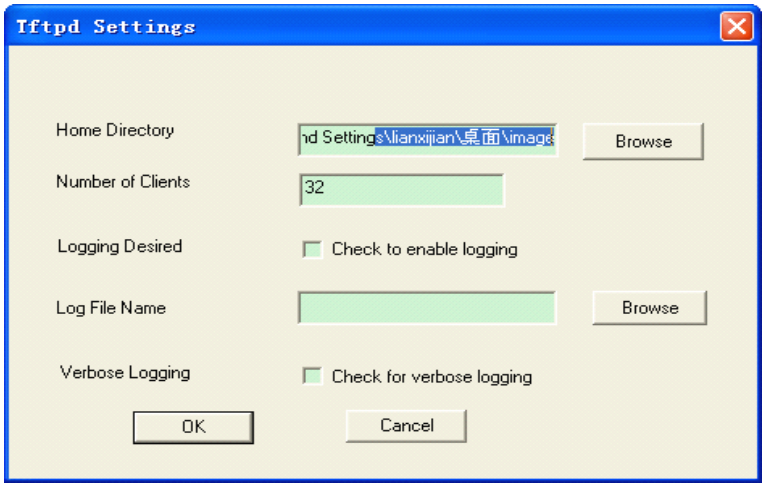



Figure 9.3.20 Tftpd.exe config interface

- Click "*Tftpd*", and click the option "*Start*" in its pull-down menu.


Figure 9.3.21 startup tftpd server

- The TFTP Server set OK.

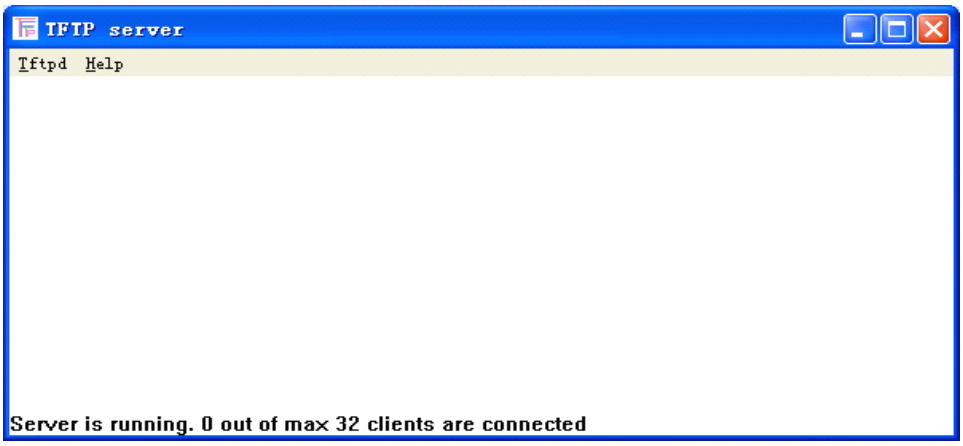



Figure 9.3.22 Tftpd server running

- Turn on the target board power, when the U-Boot enter the countdown, press the space key of the keyboard. Then U-Boot enter download image mode.

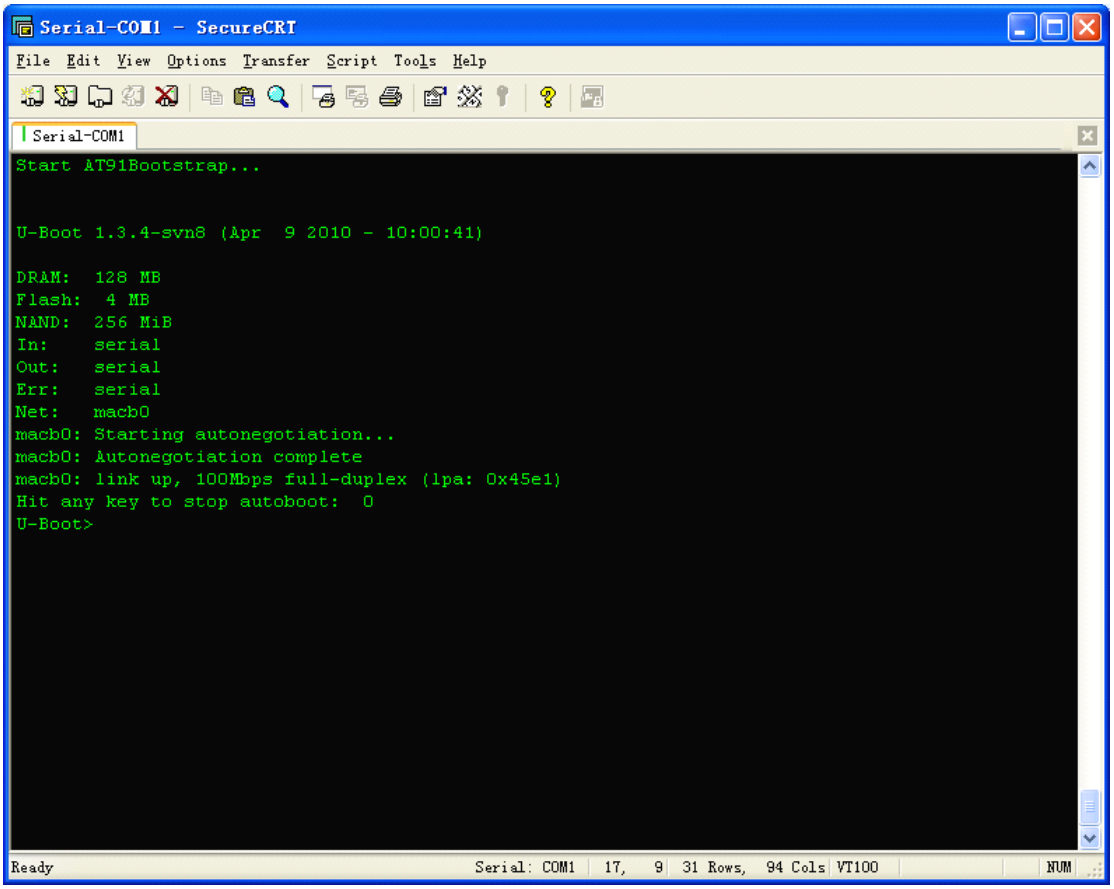



Figure 9.3.23 U-boot startup interface

- Set the target board IP and server IP, gateway IP in the U-boot

```
U-Boot> setenv serverip 192.192.192.71
U-Boot> setenv ipaddr 192.192.192.200
U-Boot> setenv ethaddr 00:11:22:33:44:55
U-Boot> saveenv
```

- Erase the nandflash space from 0x9a0000

```
U-Boot> nand erase 0x9a0000
U-Boot> nand scrub 0x9a0000
```

- Use the nand write.yaffs command to download the uImage the nandflash.

```
U-Boot> tftp 0x70000000 rfs-qtopia.yaffs2
```

- Copy the rootfs to nandflash .

```
U-Boot> nand write.yaffs 0x70000000 0x9a0000 $(filesize)
```

- The file system has been burned to the nandflash. You can reset the power.

# Chapter 10 QT Demo

You can enter the following command to start Qtopia application process.

1. Enter the following command procedures for the implementation of touch screen calibration:

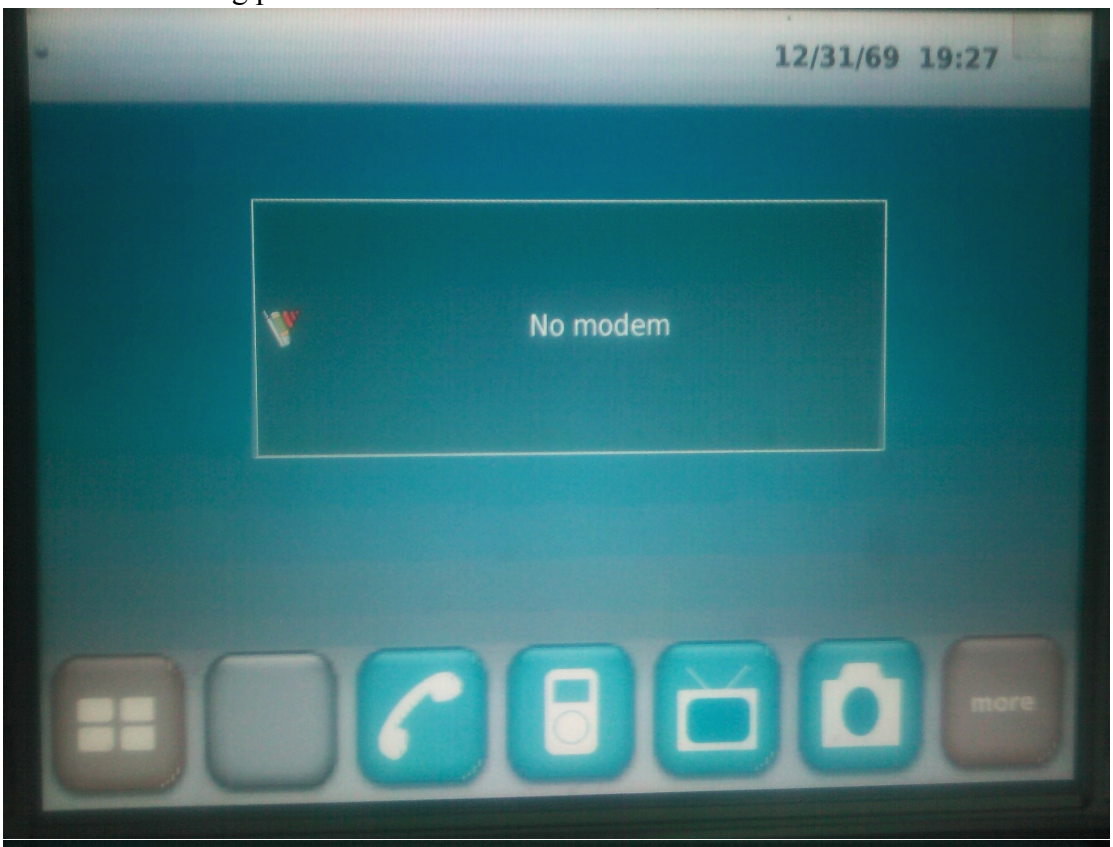
```
[root@Mini6045:/]# ts_calibrate
```

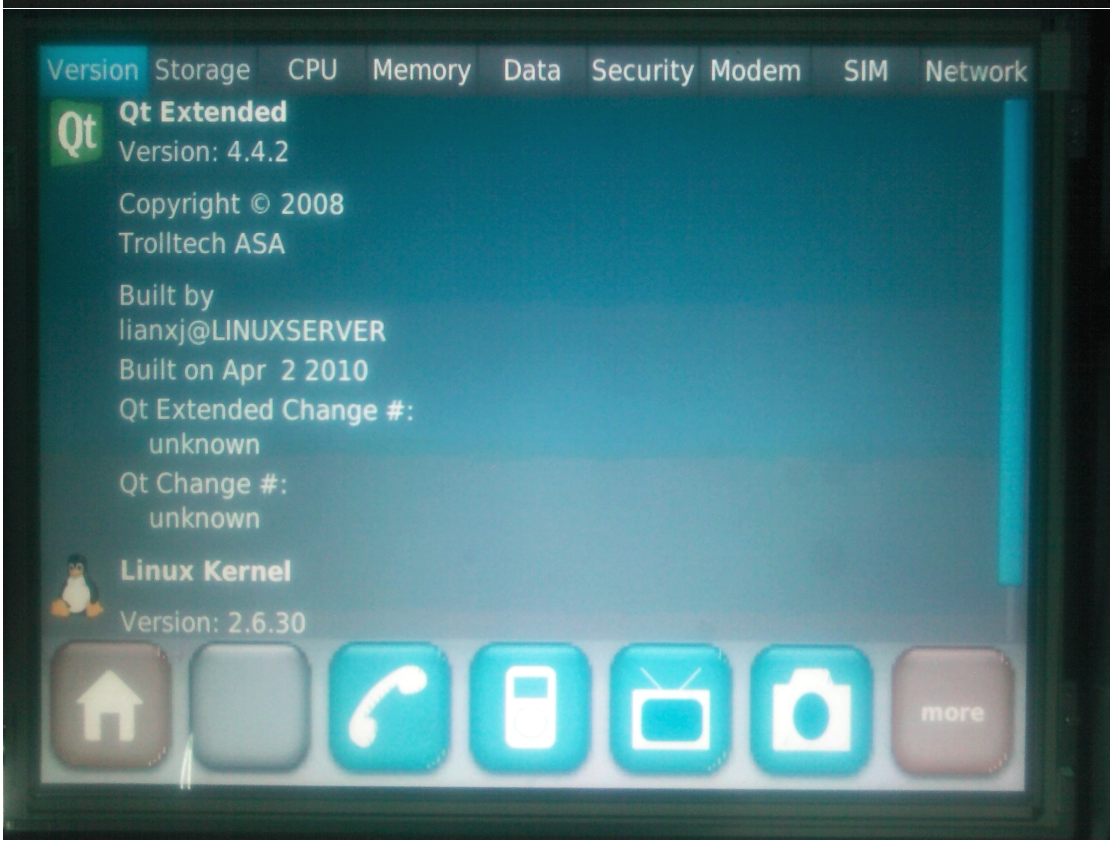
Follow the on-screen prompts, click the “+” icon five times to complete the calibration.

2. you can start the Qtopia application process with the following command.

```
[root@Mini6045:/]# qpe
```

Then the following pictures will be shown.



# Part 4 WinCE6.0 System

## Chapter 11 Introduction of WinCE6.0 System

### 11.1 Image files related to SBC6045 WinCE 6.0 system

If WinCE system is customized using SBC6045 WinCE 6.0 BSP that we provide and the compilation is successfully finished, the system will generate 6 image files, i.e.: FIRSTBOOT.nb0, EBOOT.nb0, NK.nb0, FIRSTBOOT.bin, EBOOT.bin, NK.bin, of which 4 files are usually used in burning processes, i.e.: FIRSTBOOT.nb0, EBOOT.nb0, NK.nb0, NK.bin. The files generated in .nb0 format will finally be burnt to Dataflash or NAND flash, and they can directly run in SDRAM; while the files in .bin format must first be converted into .nb0 files by EBOOT running on SBC6045 main board nb0, then they are burnt to Dataflash or NAND flash.

### 11.2 SBC6045 WinCE 6.0 Software List

Table 11.2.1 Specification of SBC6045 WinCE6.0 BSP

| Name | | Remark |
|---|---|---|
| BootLoader | Bootstrap | Boot from NAND FLASH |
| | eboot | Support programming all the images by using SAM-BA,and programming NK.bin through the network using eboot. |
| Kernel | WinCE6.0 | RAM and ROM File System<br>Hive-based Registry |
| Device Driver | Flash | Nand Flash driver |
| | Serial | CPU DEBUG and four-way Serial |
| | RS-485 | Two-way extended RS-485 |
| | CAN Bus | Two-way extended CAN BUS driver |
| | RTC | AT91SAM9G45 extended RTC driver |
| | NET | 10/100M Ethernet driver |
| | LCD | TFT LCD drive,support 4.3 inch(480x272)、5.6 inch(640X480)、7 inch（800x480）、10.4 inch（800x600）LCD. |
| | Touch Screen | The CPU touch screen control |
| | USB Host | USB Host driver |
| | USB Device | USB Device driver |
| | SD Card | SD Card driver |
| | EEPROM | 2K bits EEPROM |
| | AUDIO | Audio output |
| | LED | One controllable LED |
| | Keyboard | One custom button |
| | BEEP | One beep |

## 11.3 Principle of SBC6045 WinCE 6.0

At prsent SBC6045 WinCE 6.0 BSP Support NAND flash boot .booting flow chart as follow:

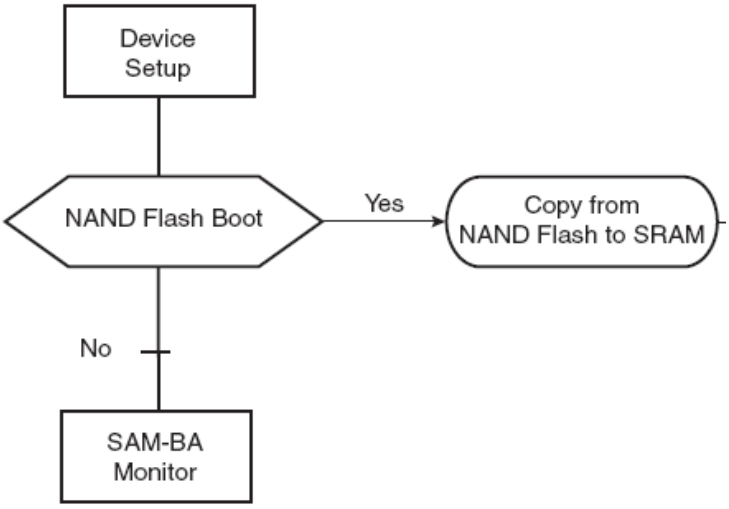



Figure11.3.1

The ROOMBOOT Code fixed inside the CPU will automatically running after power on, The ROMBOOT will check if there is a valid BOOTLOADER in nandflash. If yes the ROMBOOT programer will copy the BOOTLODER to the SRAM inside of the CPU and execute it . otherwise it will ignore the nand flash and jump to the Serial port download and excute program .

## 11.4 Principles on the booting of SBC6045 WinCE 6.0 system

After power on the system is fixed inside CPU of AT91SAM9G45, ROMBOOT will automatically copy FIRSTBOOT image, the first-level user booting code from NANDflash 0x00000000 address to SRAM inside of AT91SAM9G45 and execute it. FIRSTBOOT's role is to initialize SDRAM memory, NANDflash in AT91SAM9G45, and copy EBOOT, the second-level user booting code from NANDflash 0x00080000 address of SBC6045 main board to SDRAM of AT91SAM9G45 main board and execute it; In default state, EBOOT will automatically copy WinCE system image NK from NAND flash 0x00200000 address to SDRAM, and hand over system control to operating system. In addition, EBOOT undertakes the management operations of underlying hardware and settings of the data shared with operating system.booting flow chart as figure 11.4.1.

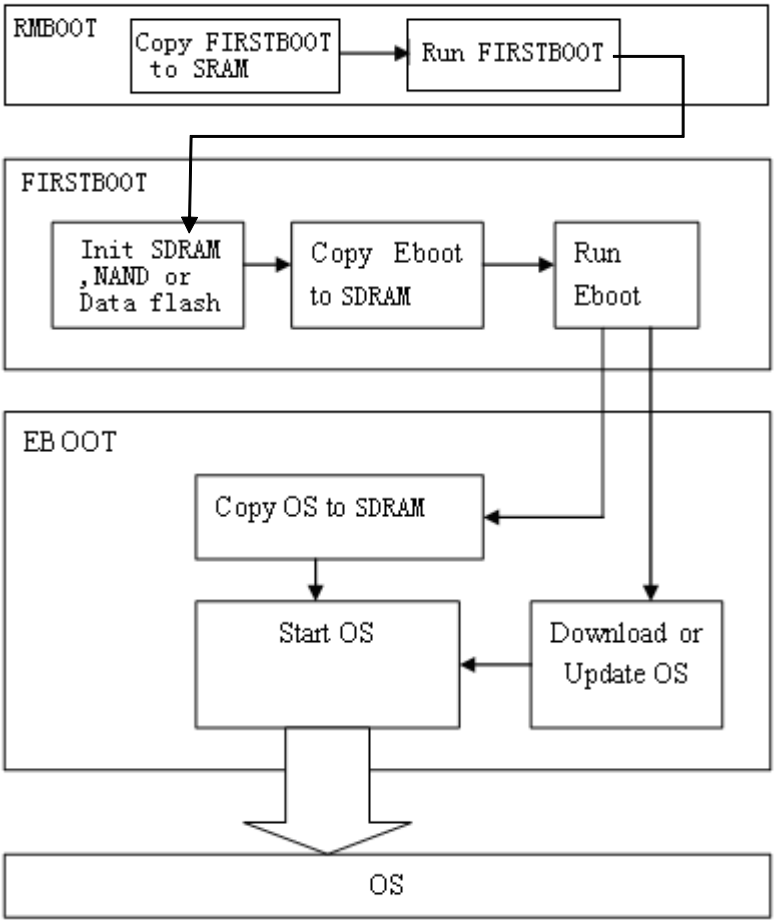



figure 11.4.1

NANDFLASH memory mapping as follow：

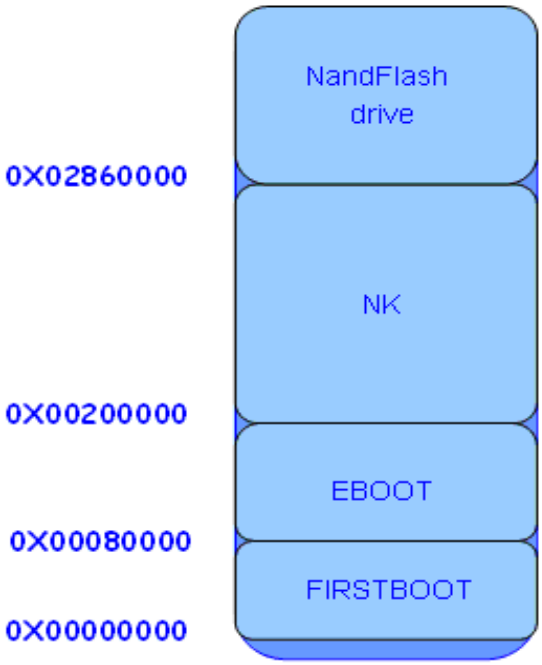



Figure 11.4.2

# Chapter 12 Burning of WinCE system

## 12.1 Introduction of burning Image of SBC6045 WinCE 6.0

Now our SBC6045 WinCE 6.0 BSP supports two burning modes, first, burn FIRSTBOOT, EBOOT and WinCE system image NK using the SAM-BA software that ATMEL provides; second, connect VS2005 WinCE 6.0 development environment installed on client development workstation or other TFTP servers (e.g.: CEDownload.exe) to SBC6045 main board, burn WinCE system image files NK via net cable using the EBOOT on SBC6045 (the EBOOT that SBC6045 WinCE 6.0 BSP of current version generates doesn't support for the burning of user booting codes FIRSTBOOT and EBOOT, it can only burn WinCE system image file NK.bin) .

Important: if you use EBOOT, the user booting code on SBC6045 main board to WinCE system image file NK, the actual process will be, EBOOT, the user booting code on SBC6045 main board will first download WinCE system image file NK.bin that VS2005 WinCE 6.0 generates from TFTP server running on development workstation to SDRAM on SBC6045 main board via net cable, then the file will be converted into NK.nb0 before being burnt to NAND flash. The TFTP server mentioned here can be the server provided in VS2005 WinCE 6.0 development environment or other TFTP servers. Please note that the port parameter of the TFTP server that you use is 980 instead of 69, the standard TFTP port. The software CEDownload.exe, which we will describe in Section 3.4 below is a TFTP server software, which has changed the default TFTP port from 69 into 980.

## 12.2 Burn SBC6045 WinCE 6.0 image through SAM-BA

### 12.2.1 Install SAM-BA software

Install the application \02 Linux 2.6 Kit\02 Tools \Install AT91-ISP v1.13.exe in SBC6045 CD on your development workstation as the default path and configurations of AT91-ISP v1.13.exe program. After the installation is finished, you will see the icon of SAM-BA v2.9 (see Figure 12.2.1) appearing on the desktop :


Figure 12.2.1

- Connect the Debug serial COM port J23.
- Connect USB device port to USB Host of the development workstation using the USB cable..
- Open the hyper terminal on the PC and setting as :115200 8n1 no CRC and no flow control.
- Before power on the SBC6045, make sure there is no image in the NAND flash .
- Double click the icon of the application SAM-BA v2.9 on desktop.


Figure 12.2.1 SAM-BA v2.9

As shown in Figure 12.2.1 to run SAM-BA 2.9 program，and select"***Select the connection***"and"***\usb\ARM0***"from the pulldown menu as shown in Figure 12.2.2.

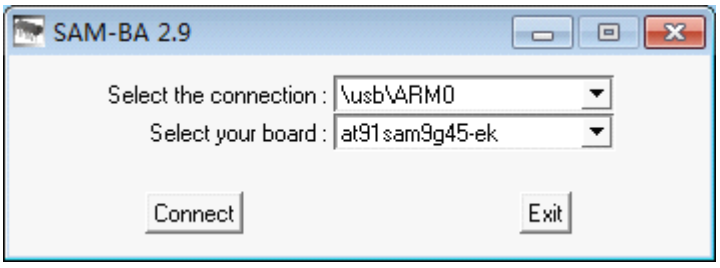



Figure 12.2.2 SAM-BA

Click "***Connect***" button as shown in Figure 12.2.2,enter the interface as show in Figure 12.2.3 below.

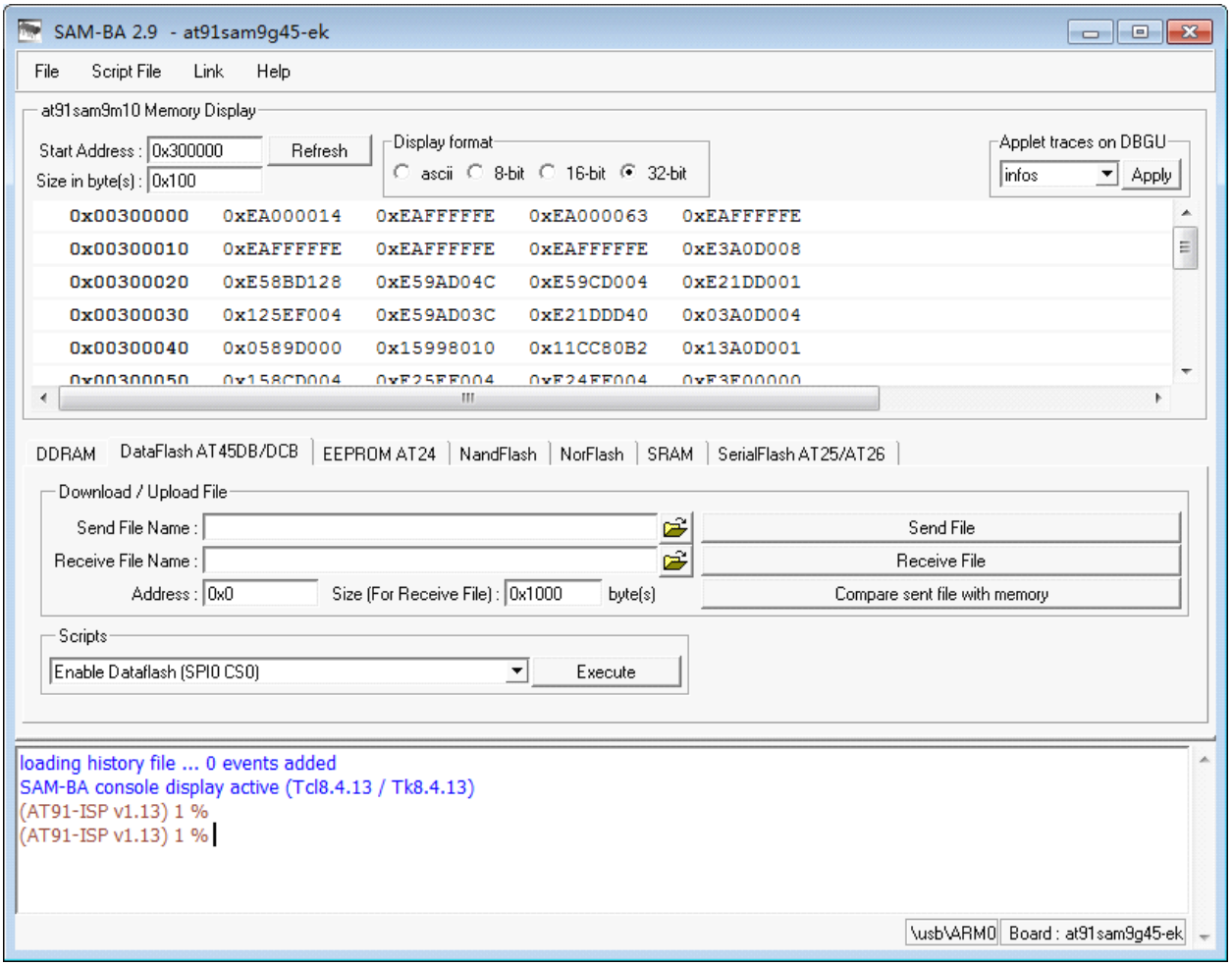



Figure 12.2.3

1. As show in Figure 12.2.4 in the Scripts pulldown menu of SAM-BA v2.9, select Enable

NandFlash, and click Execute button in the column Scripts to enable NAND flash on SBC6045 board. After the Enable operation is successful, the interface as shown in Figure 12.2.4 will appear:

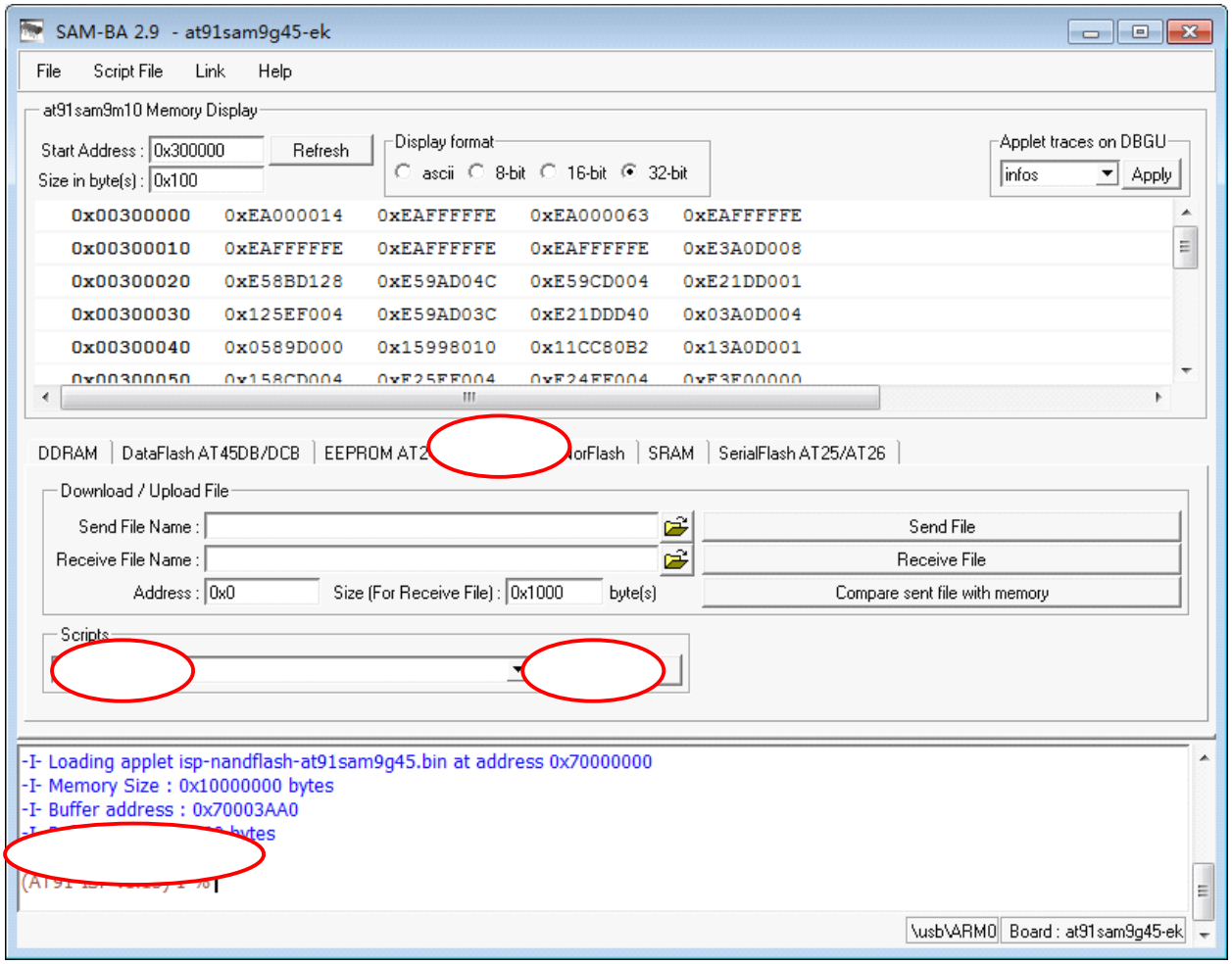



Figure 12.2.4

2. As shown in Figure 12.2.5 in the Scripts pulldown menu of SAM-BA v2.9, select Erase All, click Execute button in SAM-BA v2.9, SAM-BA v2.9 will automatically erase entire NAND flash. After the Erase operation is successful, the interface as shown in Figure 12.2.5 .

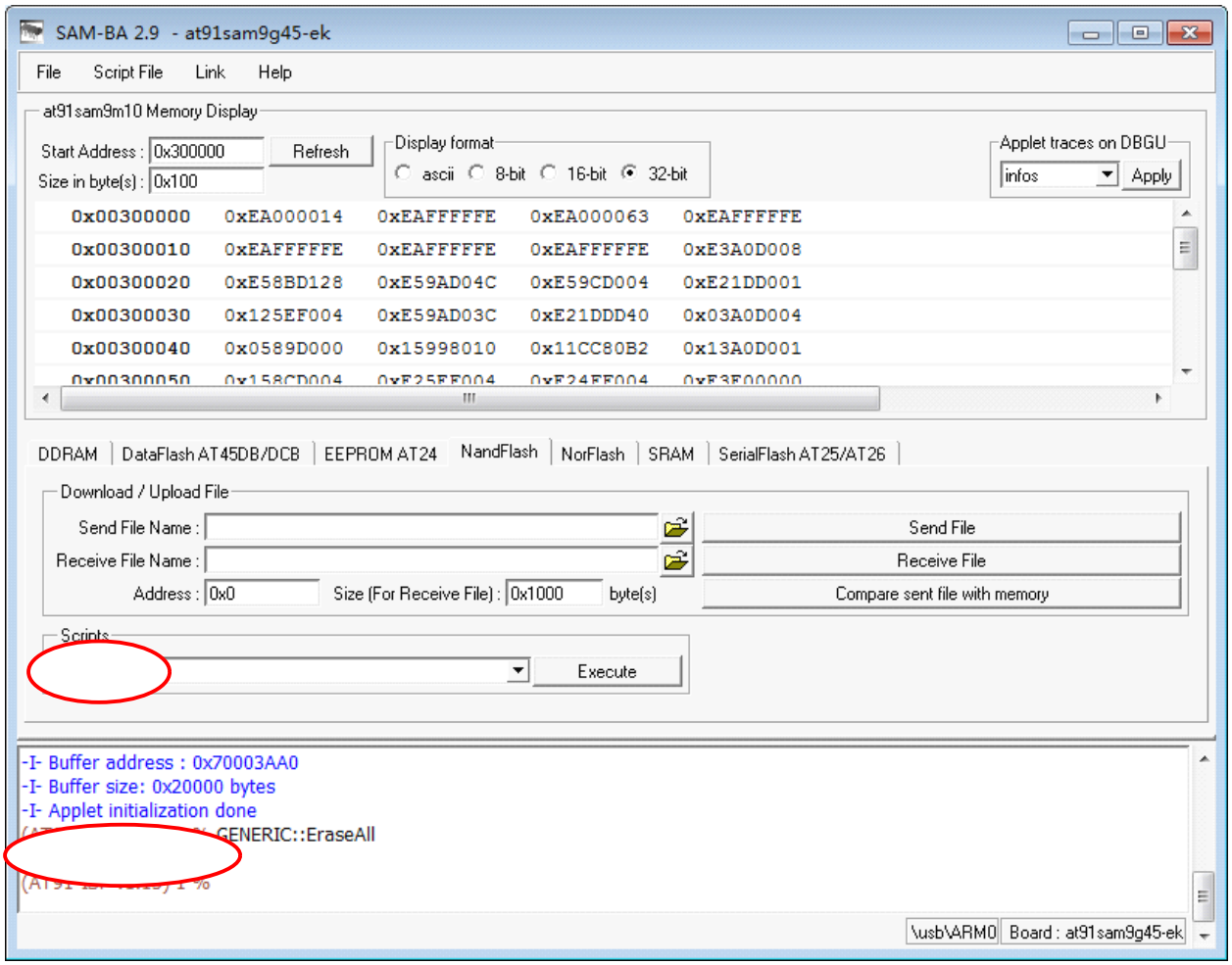



Figure 12.2.5

3. Select Send Boot File button in the Scripts pulldown menu of SAM-BA v2.9, click Execute button in SAM-BA v2.9, and open file dialog box as shown in Figure 12.2.6 will pop up in SAM-BA2.9.

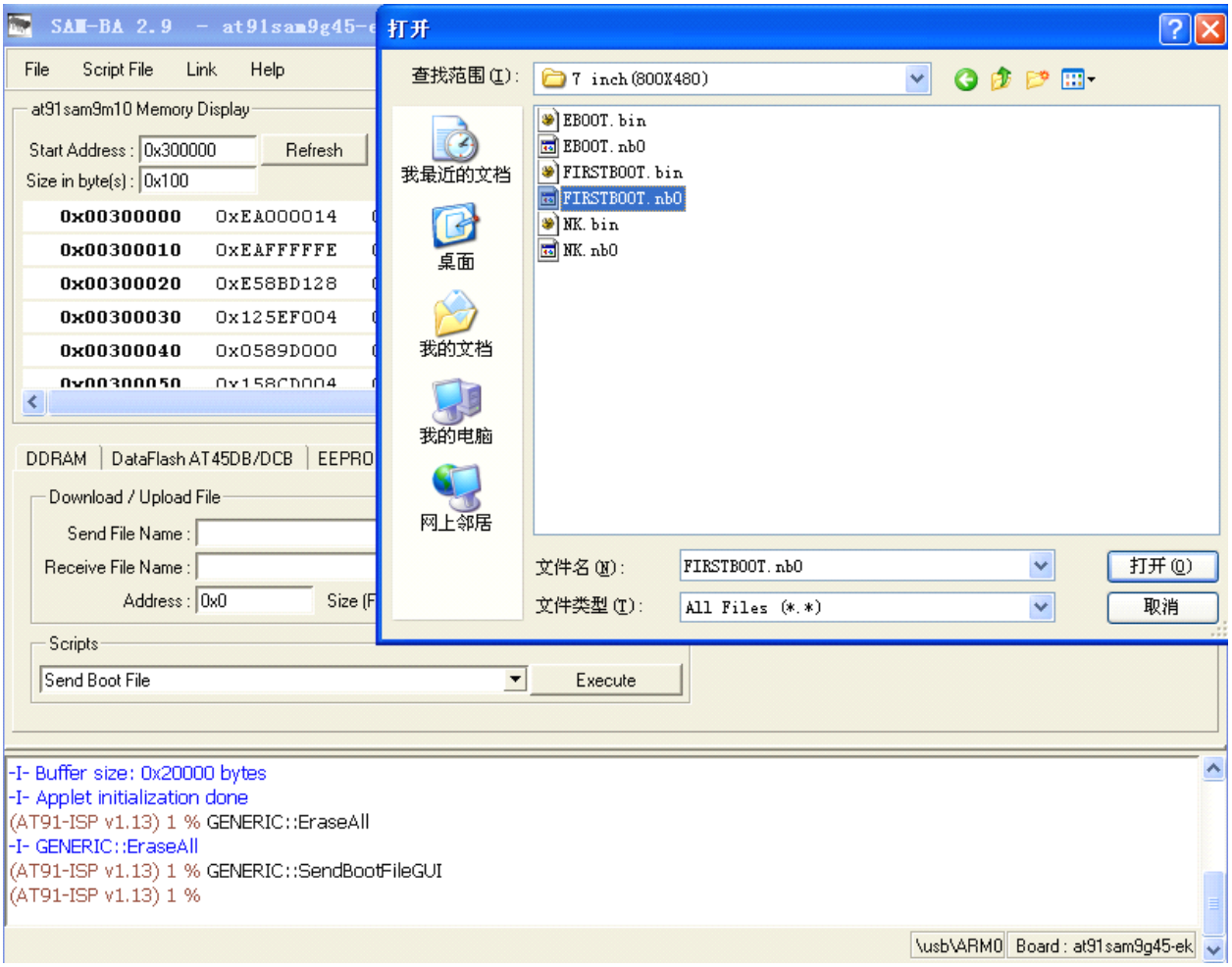



Figure 12.2.6

4. As shown in Figure 12.2.6, select FIRSTBOOT.nb0 file under the path \03 WinCE 6.0 Kit\00 Image\ in SBC6045 CD in the open file dialog box of SAM-BA v2.9, then click Open button in open file dialog box of SAM-BA v2.9, SAM-BA v2.9 will automatically start burning FIRSTBOOT.nb0 file to the beginning of 0x00000000 address of NAND flash on SBC6045 board. After the burning is successful, the interface as shown in Figure 12.2.7 will appear.

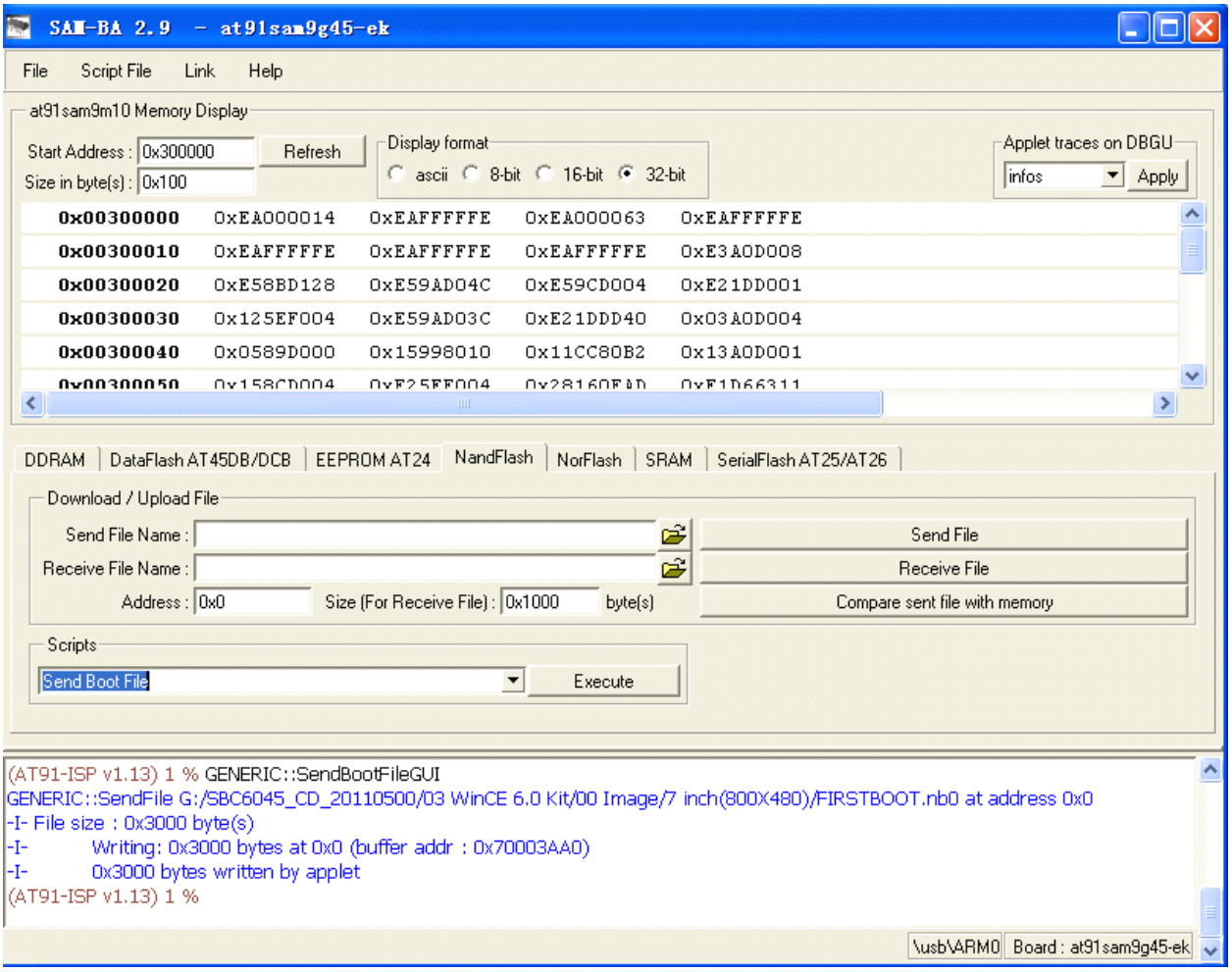



Figure 12.2.7

5. As shown in Figure 12.2.8, enter address 0x80000 in the Address: field in Download/Upload File of SAM-BA v2.9. Select SBC6045 CD \03 WinCE 6.0 Kit\00 Image \EBOOT.nb0 file in the text box Send File Name:, then click Send File button in Download/Upload File of SAM-BA v2.9, SAM-BA v2.9 will start burning EBOOT.nb0 file to the beginning of 0x80000 address of NAND flash on SBC6045 board, After the burning is successful, the interface as shown in Figure 12.2.9 will appear.

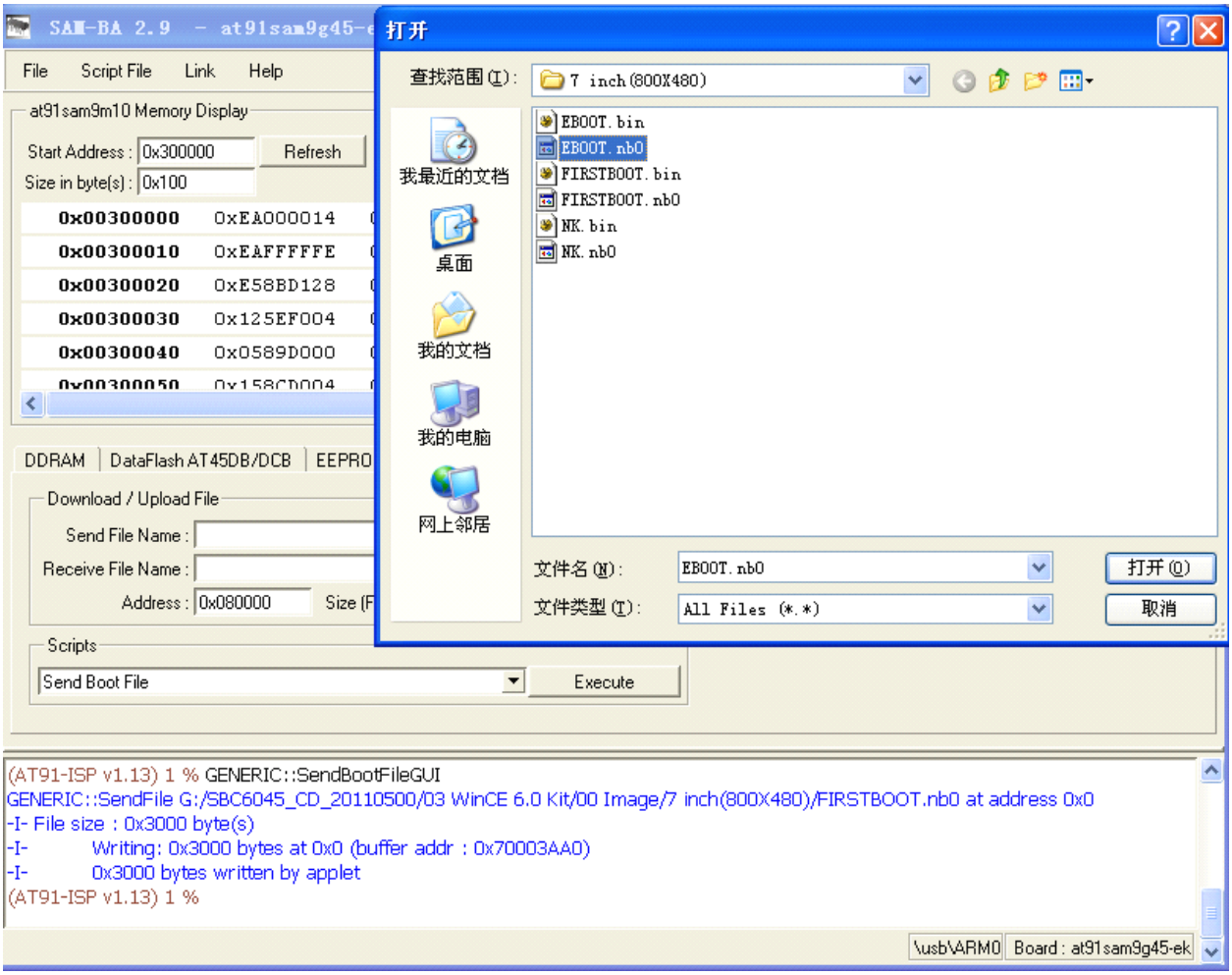

SAM-BA 2.9 - at91sam9g45-e
File
Script File
Link
Help
at91sam9m10 Memory Display
Start Address : 0x300000
Refresh
Size in byte(s) : 0x100
0x00300000 0xEA000014
0x00300010 0xEAFFFFFE
0x00300020 0xE58BD128
0x00300030 0x125EF004
0x00300040 0x0589D000
DDRAM
DataFlash AT45DB/DCB
Download / Upload File
Send File Name :
Receive File Name :
Address : 0x080000
Scripts
Send Boot File
Execute
打开
查找范围(I):
7 inch(800X480)
EBOOT.bin
EBOOT.nb0
FIRSTBOOT.bin
FIRSTBOOT.nb0
NK.bin
NK.nb0
我最近的文档
桌面
我的文档
我的电脑
网上邻居
文件名(N):
EBOOT.nb0
打开(O)
文件类型(T):
All Files (*.*)
取消
(AT91-ISP v1.13) 1 % GENERIC::SendBootFileGUI
GENERIC::SendFile G:/SBC6045_CD_20110500/03 WinCE 6.0 Kit/00 Image/7 inch(800X480)/FIRSTBOOT.nb0 at address 0x0
-I- File size : 0x3000 byte(s)
-I- Writing: 0x3000 bytes at 0x0 (buffer addr : 0x70003AA0)
-I- 0x3000 bytes written by applet
(AT91-ISP v1.13) 1 %
\usb\ARM0
Board : at91sam9g45-ek

Figure 12.2.8

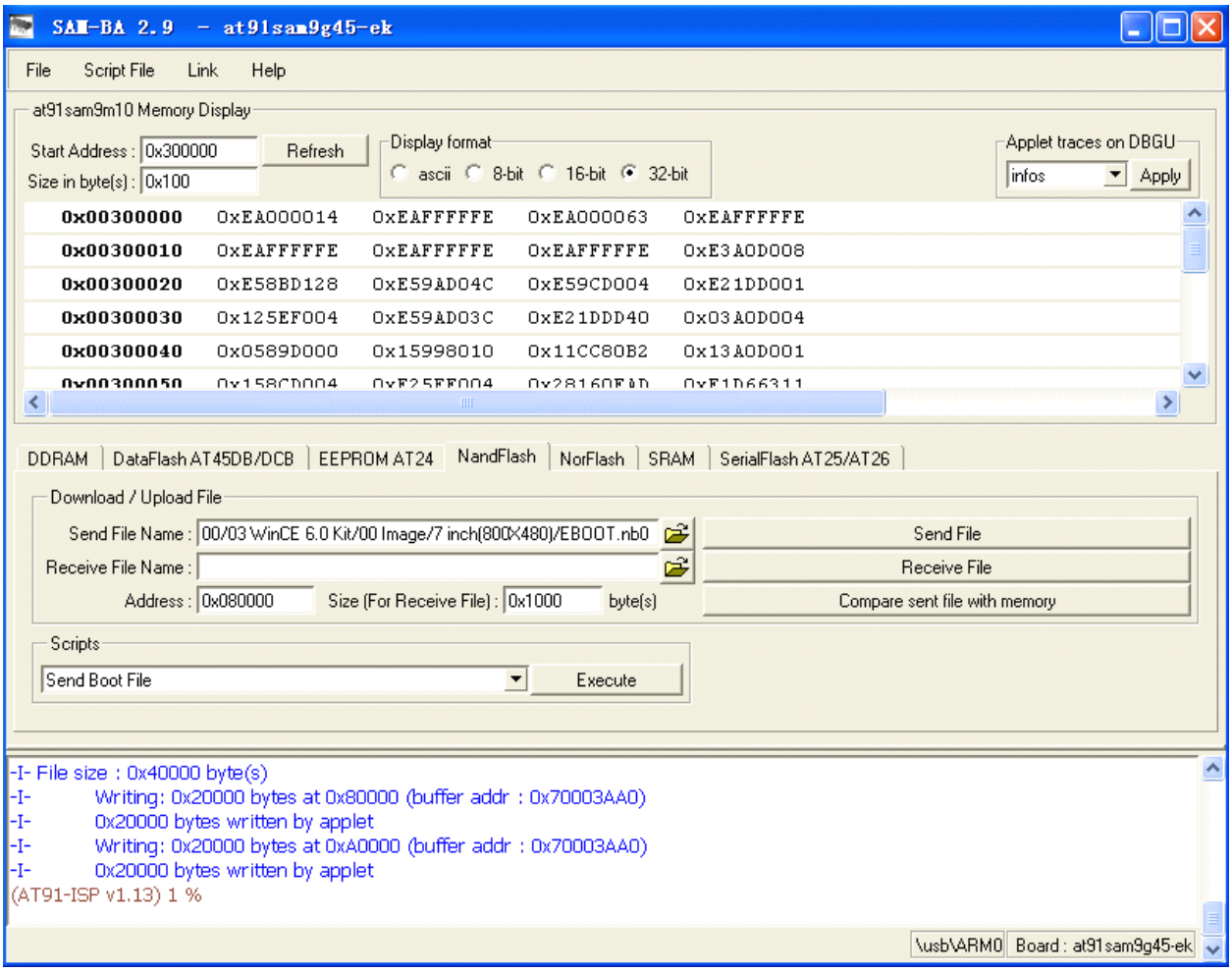



Figure 12.2.9

6. As shown in Figure 12.2.10, enter the address 0x000200000 in the Address: field in Download/Upload File of SAM-BA v2.9, select SBC6045 CD \03 WinCE 6.0 Kit\00 Image \NK.nb0 file in the text box Send File Name:, then click Send File button in Download/Upload File of SAM-BA v2.9, SAM-BA v2.9 will start burning NK.nb0 file to the beginning of 0x000200000 address of NAND flash on SBC6045 board. After the burning is successful, the interface as shown in Figure 12.2.11 will appear: (it takes 3-10 minutes to burn NK.nb0 file.) .

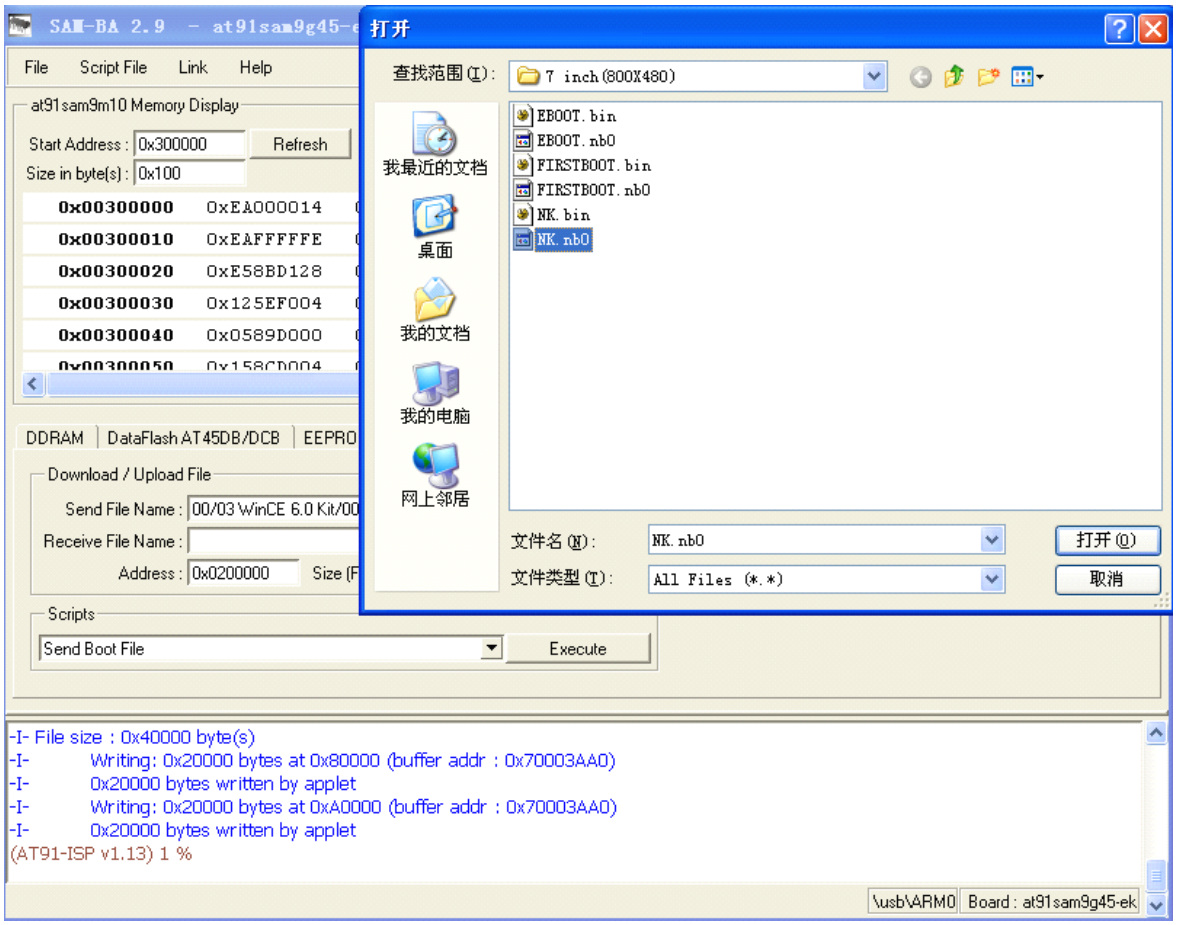



Figure 12.2.10

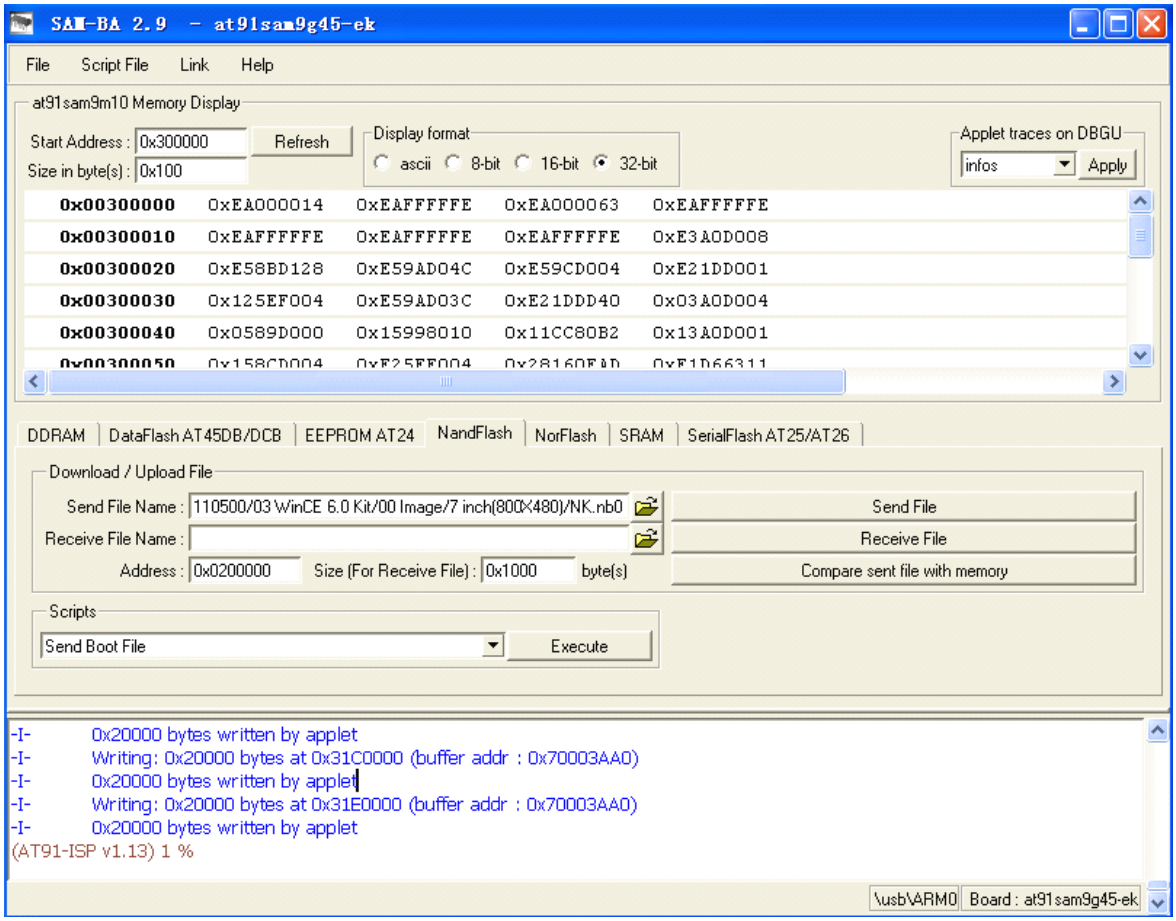



Figure 12.2.11

## 12.3 Burn WinCE 6.0 NK image files through TFTP server software CEDownload.exe

In effect, both CEDownload.exe and VS2005 WinCE 6.0 development environment as described are TFTP servers, and the default port No. of TFTP server has been changed from standard 69 into 980. This Section expounds how to update WinCE 6.0 operating system image file NK of through CEDownload.exe of SBC6045.

1. Connect Debug serial port J23 of SBC6045 to serial port of development workstation using serial port cable supplied with SBC6045. Run SBC6045 CD \03 WinCE 6.0 Kit\04 Tools \dnw.exe in development workstation,setting the communication parameter as 115200 8N1.
2. On development workstation running SBC6045 CD\03 WinCE 6.0 Kit\04 Tools\CEDownload.exe as Figure 12.3.1 show:

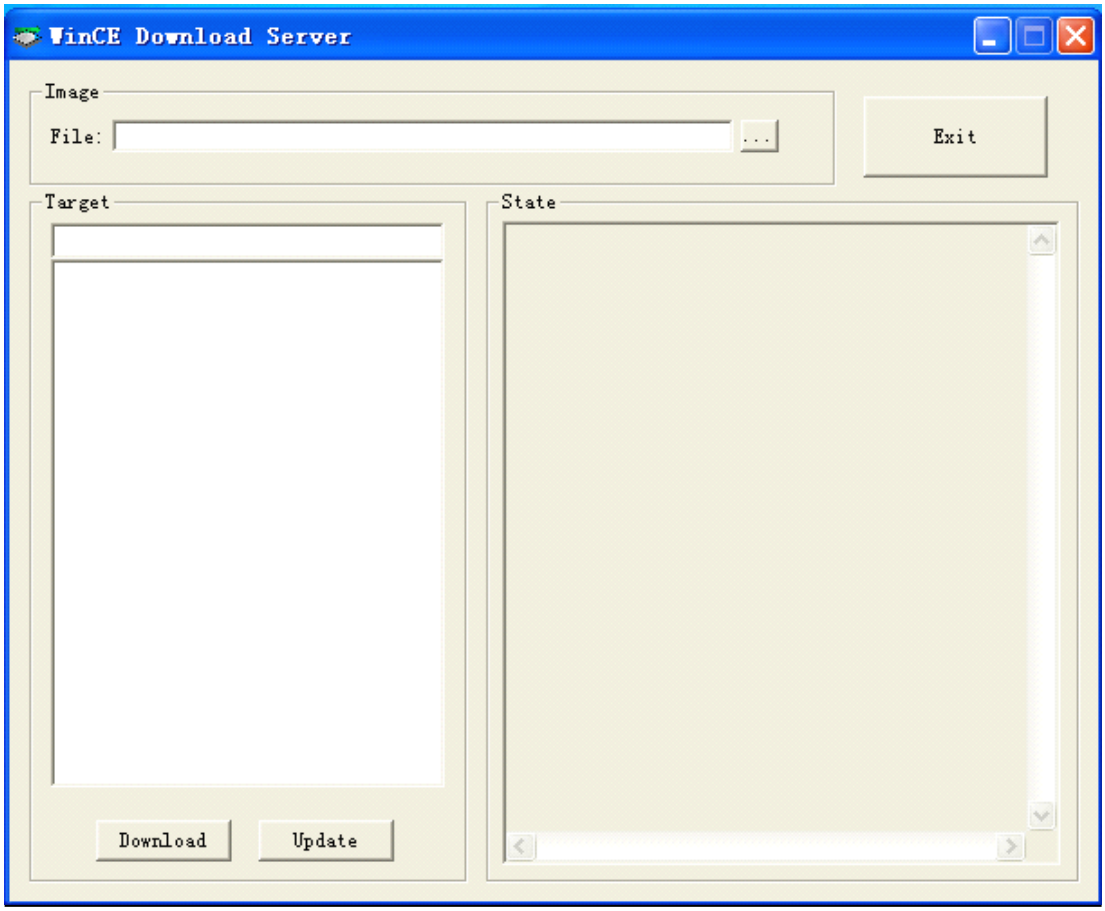



Figure 12.3.1

3. As shown in Figure 12.3.1, select NK.bin regenerated by SBC6045 WinCE 6.0 project or the NK.bin File regenerated by SBC6045 WinCE 6.0 project .Here I select NK.bin file provided in CD. As shown in Figure 12.3.2, select NK.bin file, the interface of CEDowload.exe will appear as shown in Figure 12.3.2.

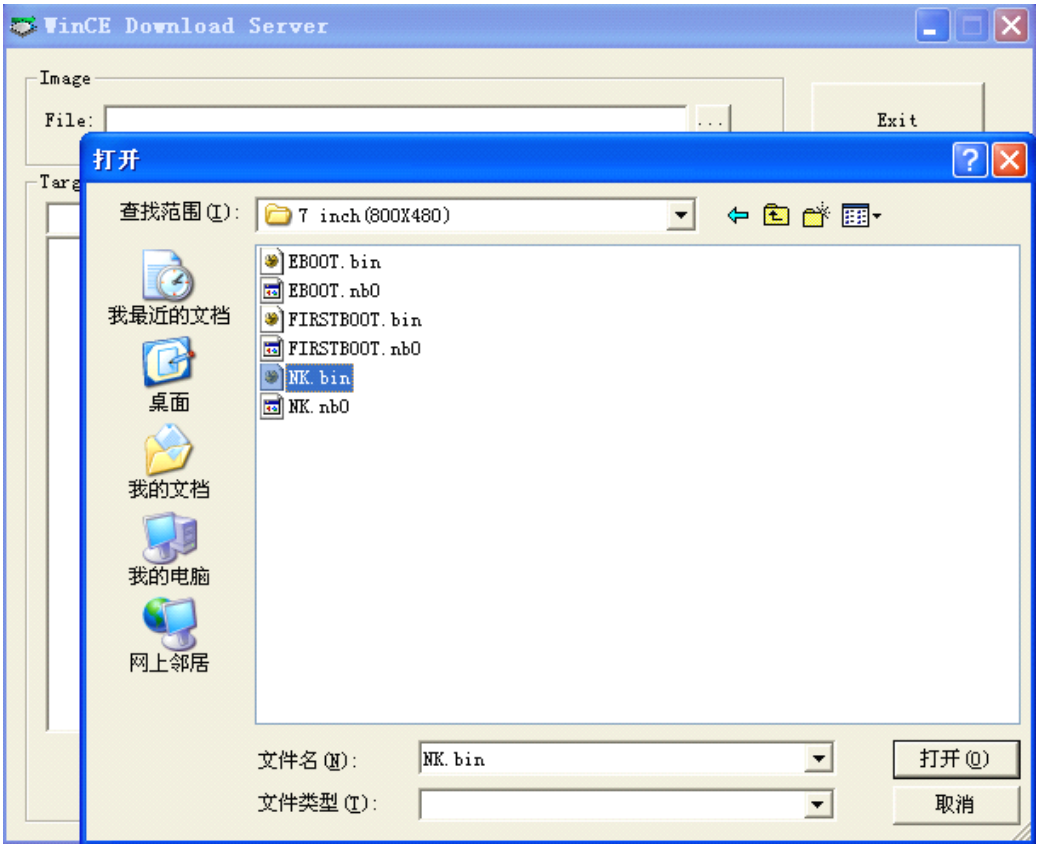



Figure 12.3.2

4. Power on SBC6045,enter space key enter EBOOT menu setting the SBC6045 config as follow figure show :

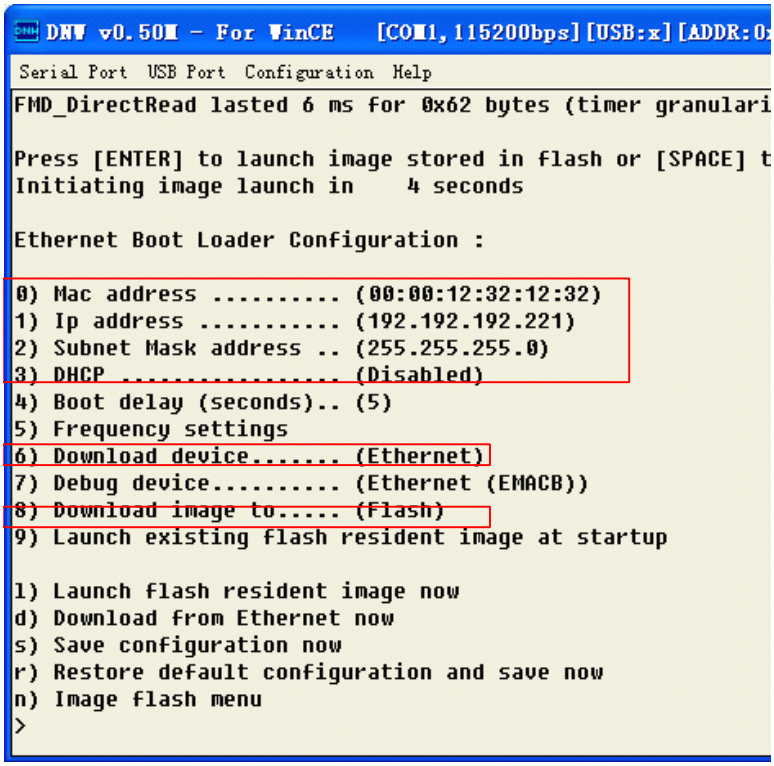



Figure 12.3.3

5. Enter the lowercase letter “d” in EBOOT menu, the system will start to download WinCE system image file NK.bin to SDRAM of SBC6045 via net cable, you will see the SBC6045 serial port output information as shown in Figure 12.3.4 will appear in DNW tool.

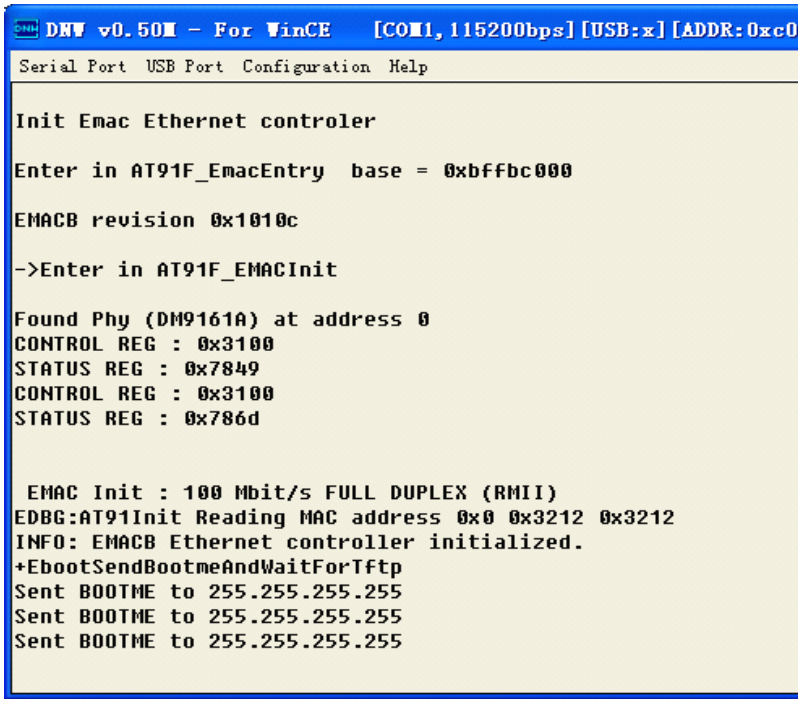



Figure 12.3.4

6. When you see the following information in DNW Sent BOOTME to 255.255.255.255 As shown in Figure 12.3.4, press Download button in CEDownload.exe software, EBOOT of SBC6045 will start to download NK.bin file to SDRAM via net cable, after the downloading is finished, it will first convert NK.bin file into NK.nb0 file, then burn it into NAND flash, in the end, it will boot WinCE system. Figure 12.3.5 shows the downloading progress appearing in CEDownload software .

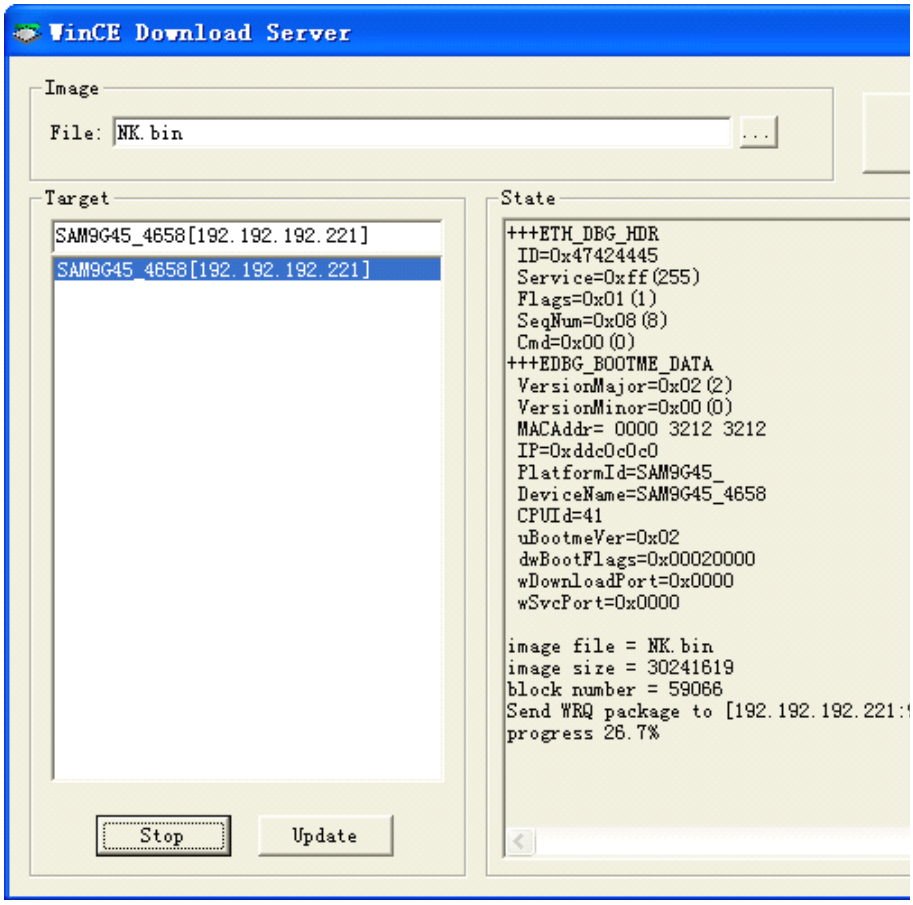



Figure 12.3.5

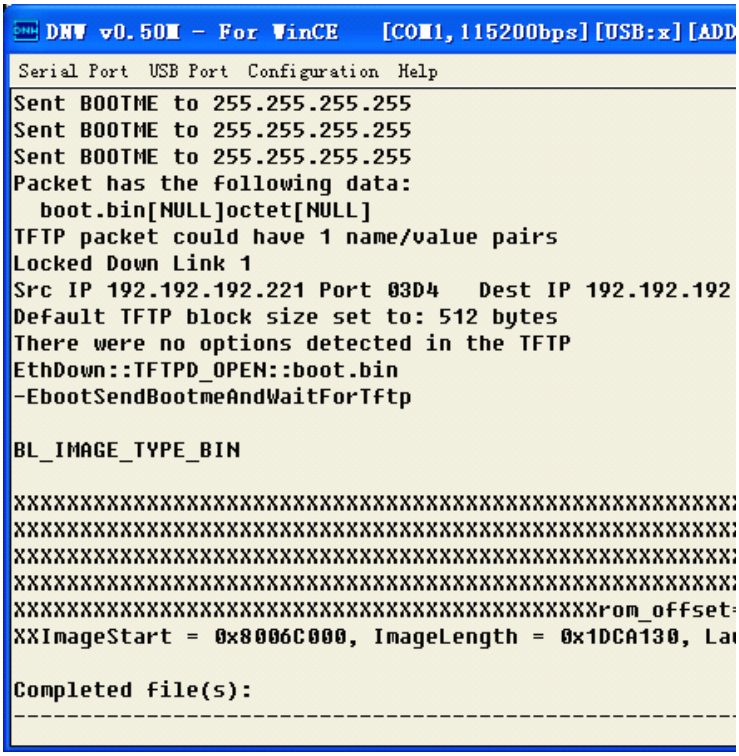



Figure 12.3.6

7. After the downloading progress is completed,the system will automatically run, as shown in Figure 12.3.7.

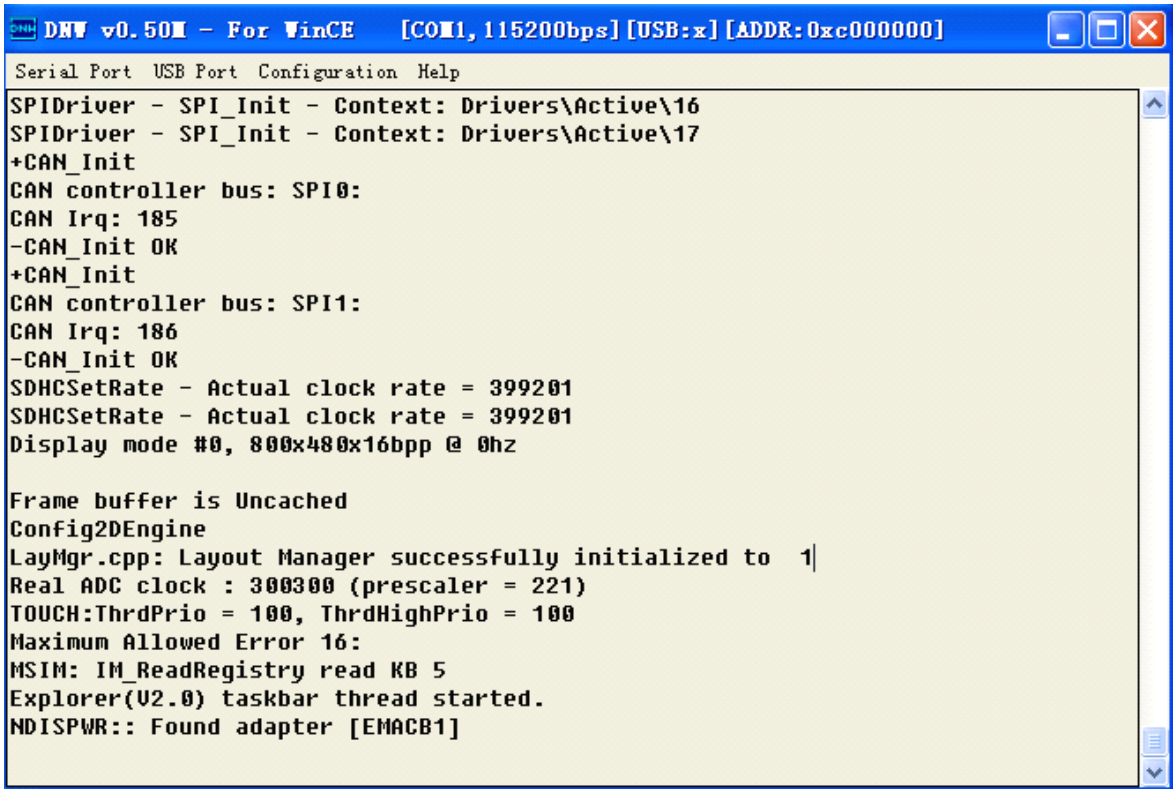



Figure 12.3.7

# Chapter 13 Customization of WinCE system based on SBC6045 WinCE 6.0 BSP

*Notes:*
*To develop Windows Embedded CE 6.0 operating system using SBC6045 BSP, you need setup Windows Embedded CE 6.0 development workstation.*
*This Manual specifies that Windows Embedded CE 6.0 development workstation software is to be installed in Driver D, that is to say, the installation path for Windows Embedded CE 6.0 is [D:\WINCE600].*

## 13.1 Installation of SBC6045 WinCE 6.0 BSP

Unzip SBC6045 CD \03 WinCE 6.0 Kit\01 BSP \SBC6045.rar to D:\WINCE600\PLATFORM of development workstation. Unzip SBC6045 CD \03 WinCE 6.0 Kit\01 BSP \ATMEL.rar to D:\WINCE600\PLATFORM\COMMON\SRC\SOC of development workstation, then the installation of SBC6045 WinCE 6.0 BSP is finished.

## 13.2 Modules of SBC6045 WinCE 6.0 BSP

SBC6045 WinCE 6.0 BSP consists of two parts, i.e.: the two folders, SBC6045 and ATAMEL, the

SBC6045 folder stores the board-level codes of SBC6045 and WinCE registry files and other WinCE configuration files. Here the code files is used to configure pin functions, configuration files are used to build WinCE system. The ATMEL folder stores AT91SAM9G45 chip-level codes. In general, clients don't have to modify the code under ATAMEL folder, they only need modify code and configuration files under SBC6045 folder.

The following paragraphs will describe the folders and files under the SBC6045 folder. In general, the codes under the ATMEL folder are in no need to change, so the Manual doesn't provide details of these codes. If you need change them, please view the files and folders under the ATMEL folder by yourself.

The directory tree of SBC6045 folder and the definitions of the folders and files under it are shown as below:

```
├─CATALOG        This directory stores SBC6045.pbcxml, this file is used to view the modules in VS2005 Catalog Items View
├─SRC
│  ├─BOOTLOADER
│  │  ├─EBOOT          This directory stores public codes of EBOOT
│  │  ├─FirstBoot      This directory stores the first-level user booting code
│  │  ├─SplashScreen   code be interrelated to screen display
│  │  ├─SDCard         This directory store the code that WinCE system Eboot uses when booting from SD card
│  │  └─SPIDataFlash   This directory stores the codes that WinCE system EBOOT uses when booting from Dataflash
│  ├─DRIVERS
│  │  ├─Display        This directory stores pin configure files of SBC6045 TFT LCD driver
│  │  ├─GPIO           This directory stores SBC6045 AT91SAM9G45 chip-level IO driver code
│  │  ├─I2C            This directory stores driver code of SBC6045 AT9G45 chip I2c interface
│  │  ├─KeyPad         This directory stores driver code of pin keys of SBC6045 AT91SAM9G45 chip
│  │  ├─NandFlash      This directory stores NAND flash driver code
│  │  ├─EEPROM         This directory stores eeprom driver code
│  │  ├─SDHC           This directory stores SD card slot SDIO mode driver code of SBC6045 board
│  │  ├─Serial         This directory stores driver code of serial port outgoing from AT91SAM9G45 chip
│  │  ├─EMACB          This directory stores dirvers of AT91SAM9G45 embeded MAC
│  │  ├─EmacbNDIS      This directory stores net NDIS driver
│  │  ├─SPI            This directory stores driver code of SPI pin outgoing from AT91SAM9G45 chip
│  │  ├─CAN            This directory stores driver code of CAN Bus
│  │  ├─RTC            This directory stores driver code of external RTC chip DS3231
│  │  ├─WAVEDEVWM9711  This directory stores driver code of WM9711
│  │  ├─WAVEDEVAD1981  This directory stores driver code of AD1981 Audio chip
│  │  ├─VideoDecoder   This directory stores driver code of video decoder
│  │  ├─TouchScreen    This directory stores driver code of touch screen
│  │  ├─USBHSFN        This directory stores driver code of USB device pin outgoing from AT91SAM9G45 chip
│  │  └─USB_Host       This directory stores driver code of USB Host pin outgoing from AT91SAM9G45 chip
│  ├─INC               This directory stores the head files to be used in BSP
│  ├─KITL              This directory stores the code related to KITL
│  ├─MISC
```

```
|   └─OAL
|       ├─OALEXE    generates the code related to WinCE 6.0 OAL.EXE
|       └─OALLIB    generates the code related to WinCE 6.0 OAL.EXE
└─FILES             This directory stores configure files will used in customer system.
```

## 13.3 Compilation of WinCE system based on SBC6045 WinCE 6.0 BSP

Here we will not describe how to customize a WinCE project based a BSP in VS2005 WinCE 6.0 development environment, user can directly copy SBC6045 CD \03 WinCE 6.0 Kit\02 Project \SBC6045.rar file to the directory D:\WINCE600\OSDesigns\ of development workstation, and unzip this file under the same directory. User can directly open the project file SBC6045.sln [under D:\WINCE600\OSDesigns\SBC6045 folder], then you can select [Build-> Build Solution] in VS2005 to start compiling the customized WinCE system.

## 13.4 Compilation of WinCE system based on SBC6045 WinCE 6.0 BSP

This section will describe the steps on how to customize WinCE 6.0 in SBC6045 BSP:

1. Open Visual Studio 2005, click File->New->Project, see Figure 13.4.1 below:

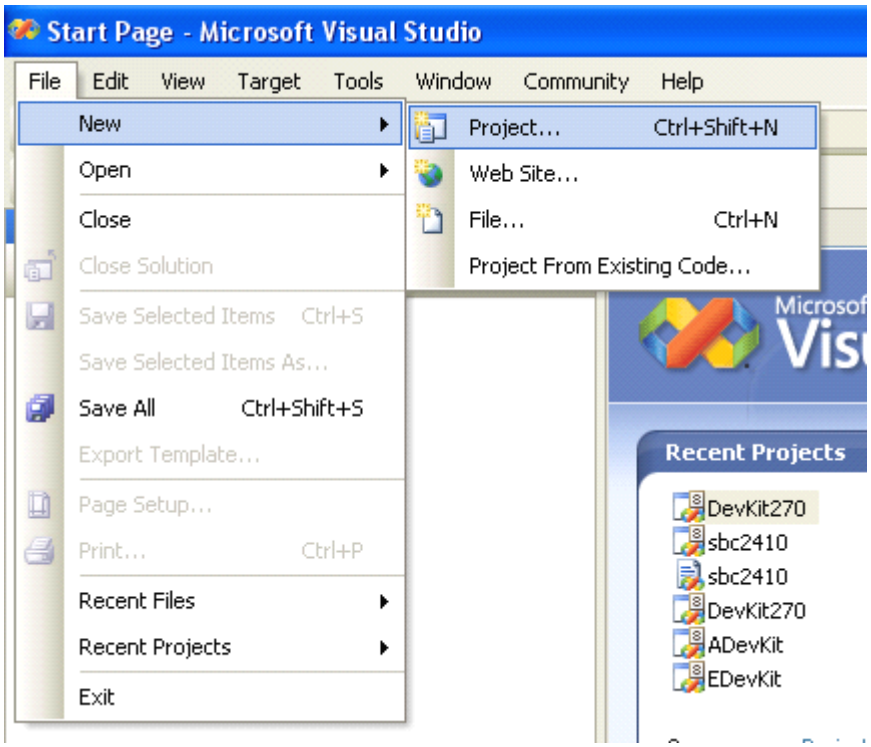



Figure 13.4.1

2. Select Platform Builder for CE 6.0 from Other project types in New Project wizard, enter SBC6045 in name:, see Figure 13.4.2:


Figure 13.4.2

3. Click OK button as shown in Figure 13.4.2, the system will next step of Windows Embedded CE 6.0 OS Design Wizard, see Figure 13.4.3:

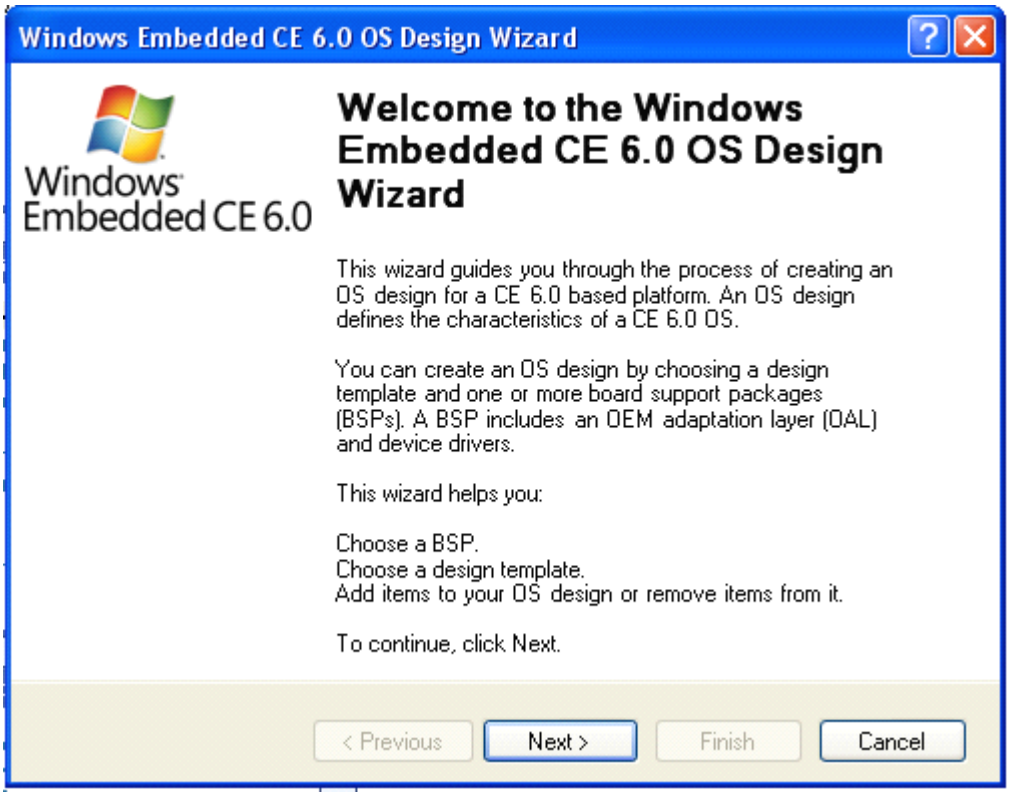



Figure 13.4.3

4. Click Next button in Windows Embedded CE 6.0 OS Design Wizard as shown in Figure 13.4.3, select SBC6045:ARMV4I in Board Support Packages (BSPs) as shown in Figure 13.4.4:

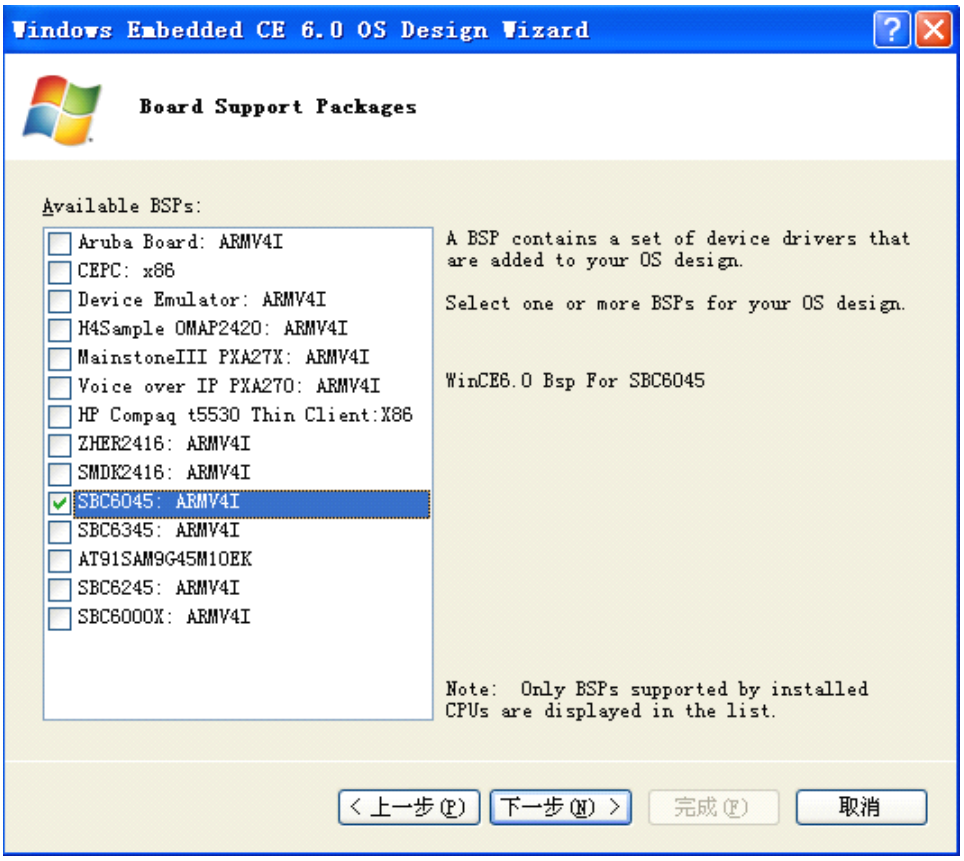



Figure 13.4.4

5. Click Next button as shown in Figure 13.4.4, select PDA Device in newly opened Design Templates->Available design templates, see Figure 13.4.5:

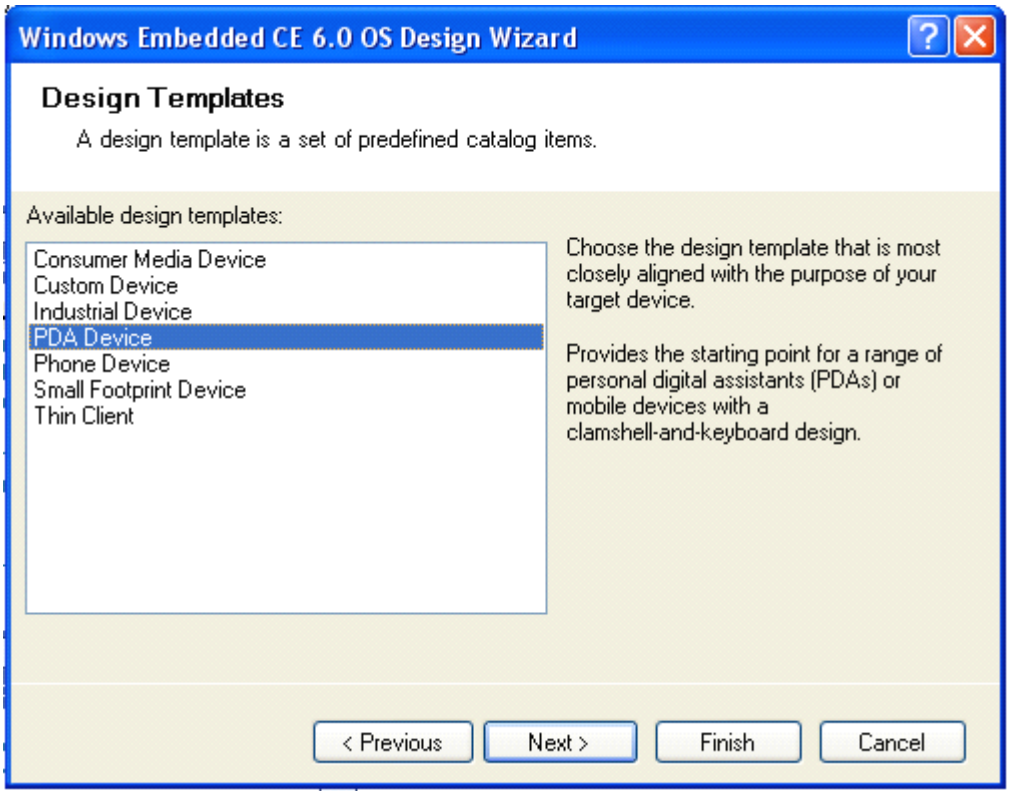



Figure 13.4.5

6. Click Next as shown in Figure 13.4.5, select Mobile Handled in newly opened Design Template Variants->Variants, see Figure13.4.6:

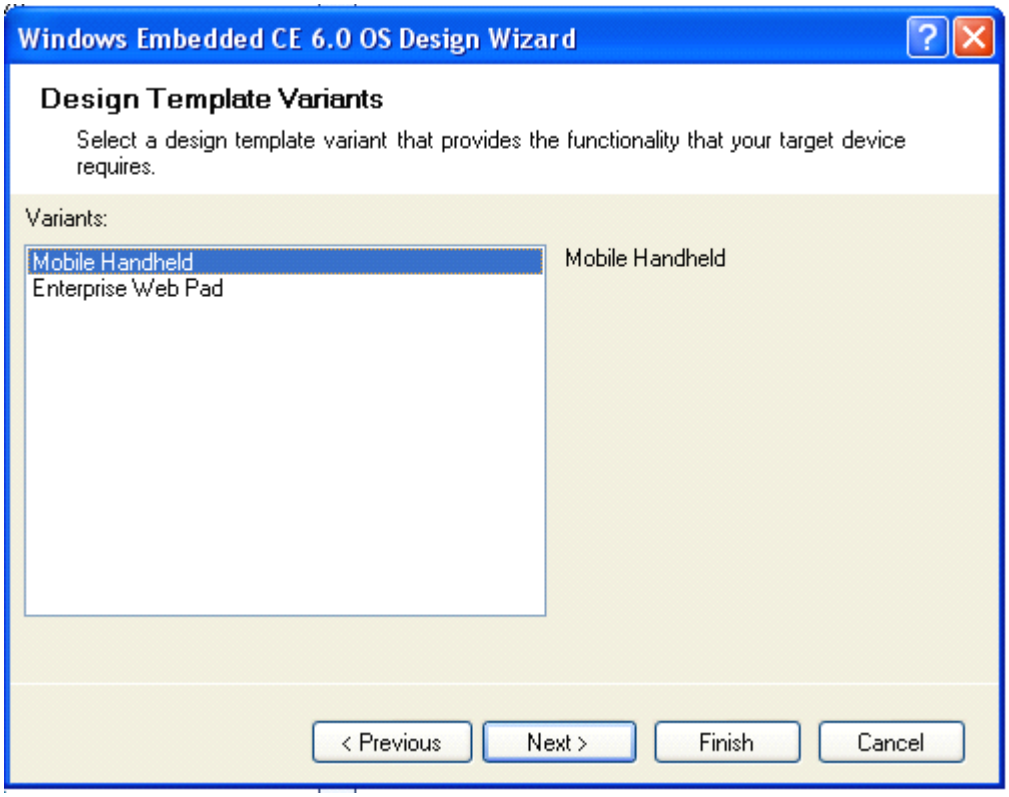



Figure 13.4.6

7. Click Next button as shown in Figure 13.4.6 to open Applications & Media window, add Internet Browser->Internet Explorer 6.0 and WordPad in existing default options, see Figure 13.4.7

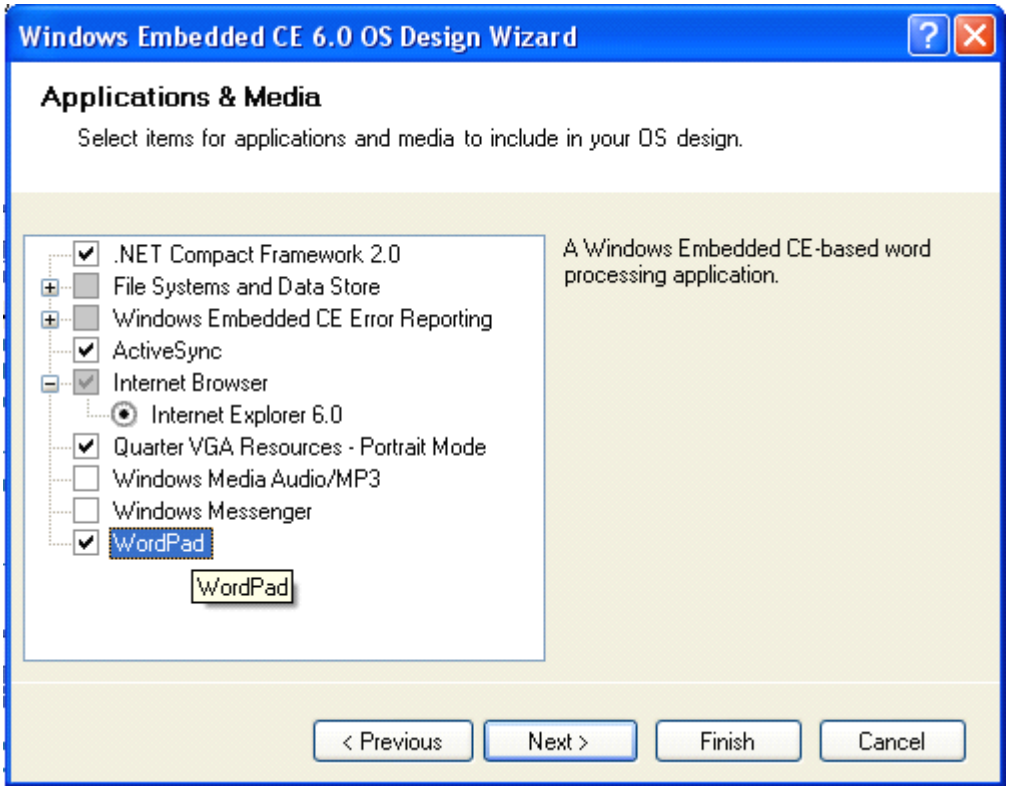



Figure 13.4.7

8. Click Next button as shown in Figure 13.4.7 to open Networking & Communications, remove Personal Area Network(PAN)->Bluetooth and Personal Area Network(PAN)->IraDA from in existing default options, see Figure13.4.8:

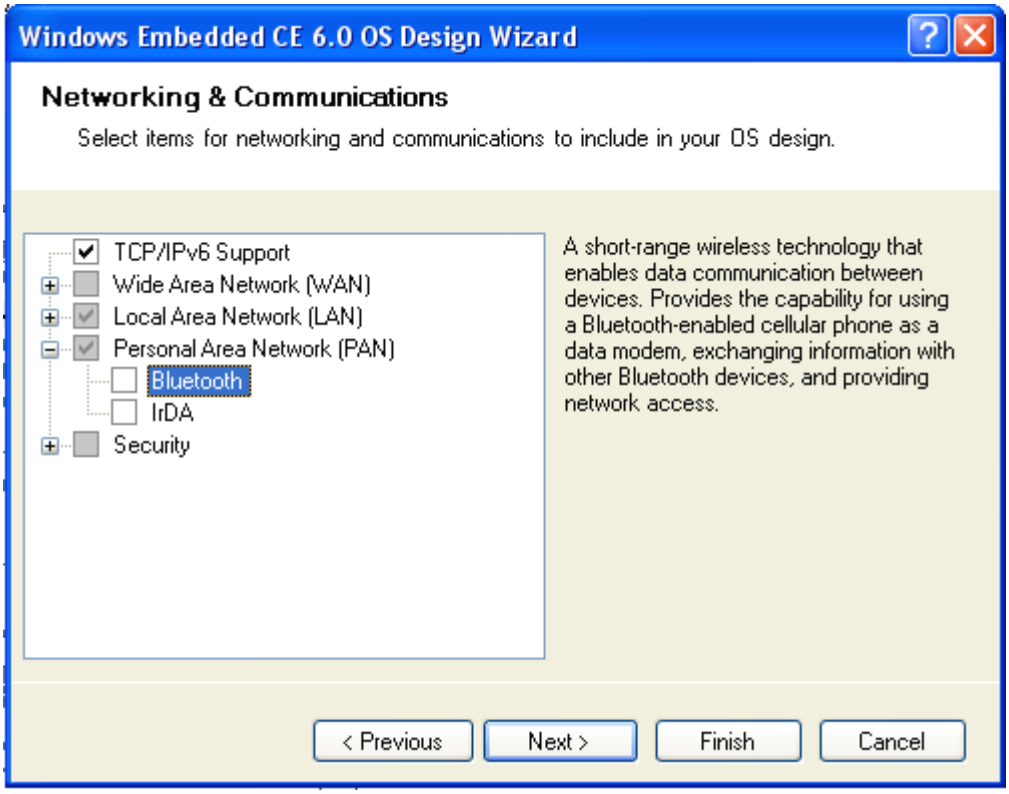



Figure 13.4.8

9. Click Next button as shown in Figure 13.4.8, Click finish in newly opened OS Design Project Wizard Complete, see Figure 13.4.9:

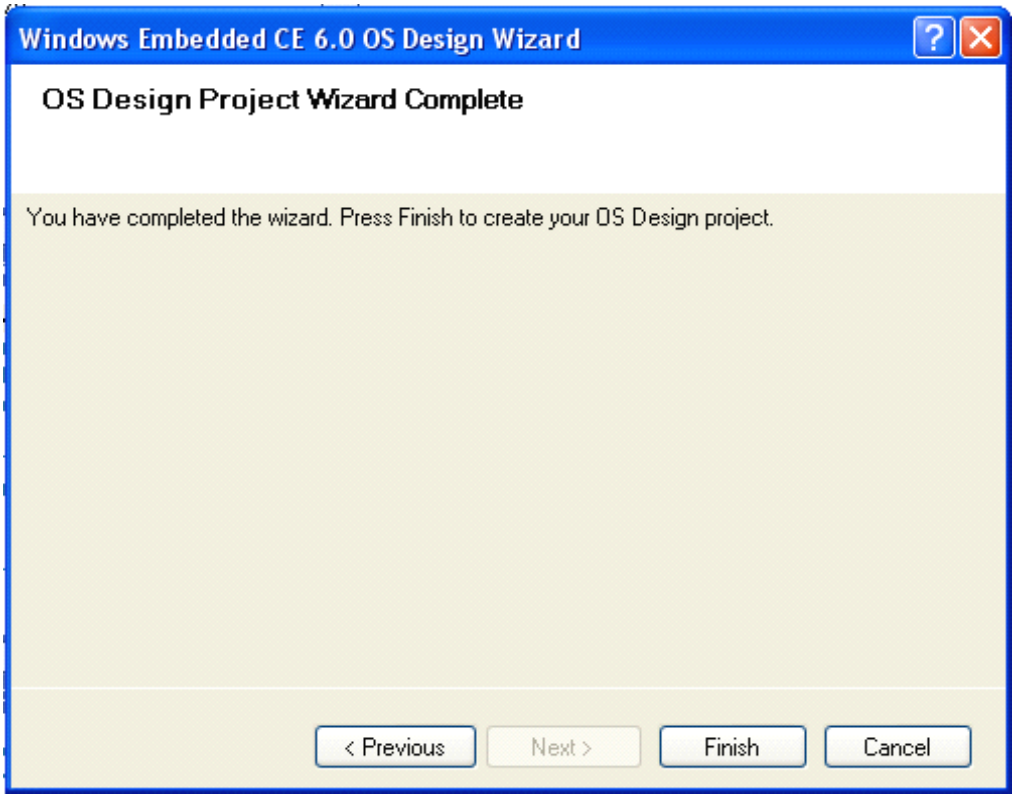



Figure13.4.9

10. Click Acknowledge in the interface as shown in Figure 13.4.10 to finish the initial customization of WinCE 6.0 OS.

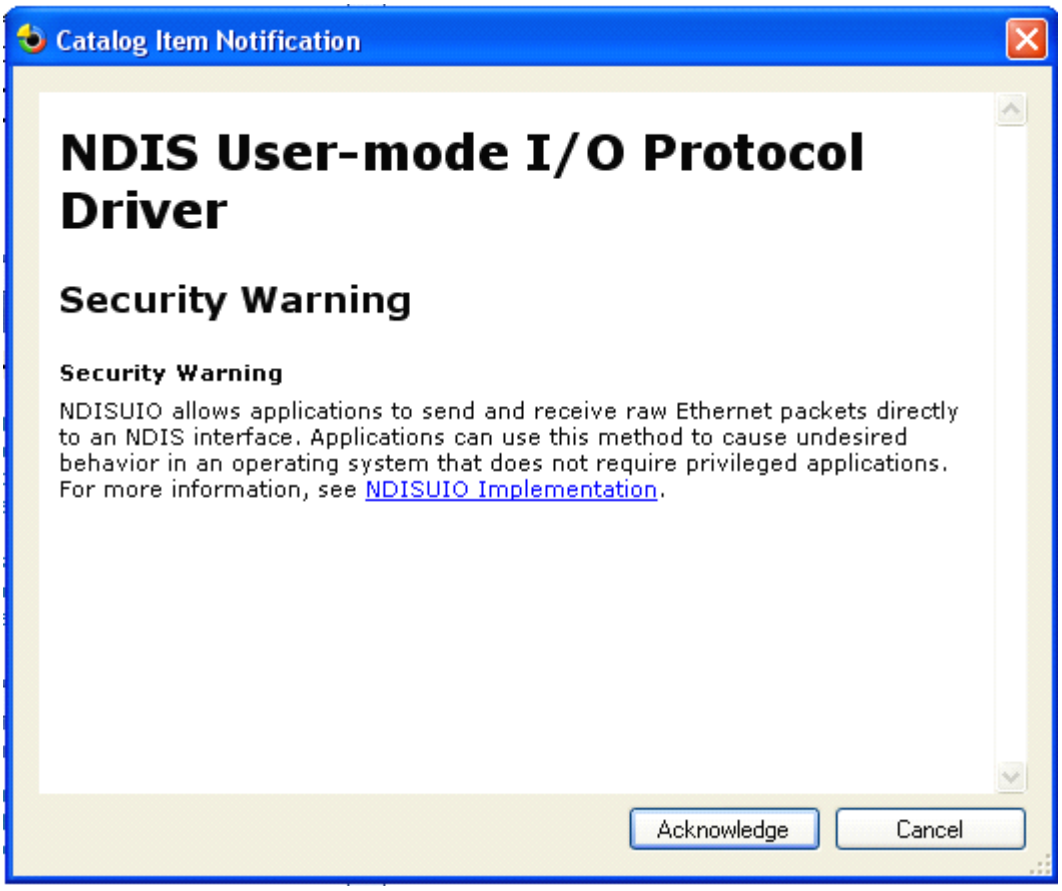


Figure13.4.10

11. Tick the following options under View->other windows->Catalog Items View->SBC6045->Core OS->CEBASE->Core OS Services->USB HOST Support in Catalog Items View of VS2005.

| |
|---|
| USB Function Driver<br>USB Host Support<br>USB Human Input Device (HID) Class Driver->USB HID Keyborad and Mouse<br>USB Storage Class Driver |

See as Figure 13.4.11

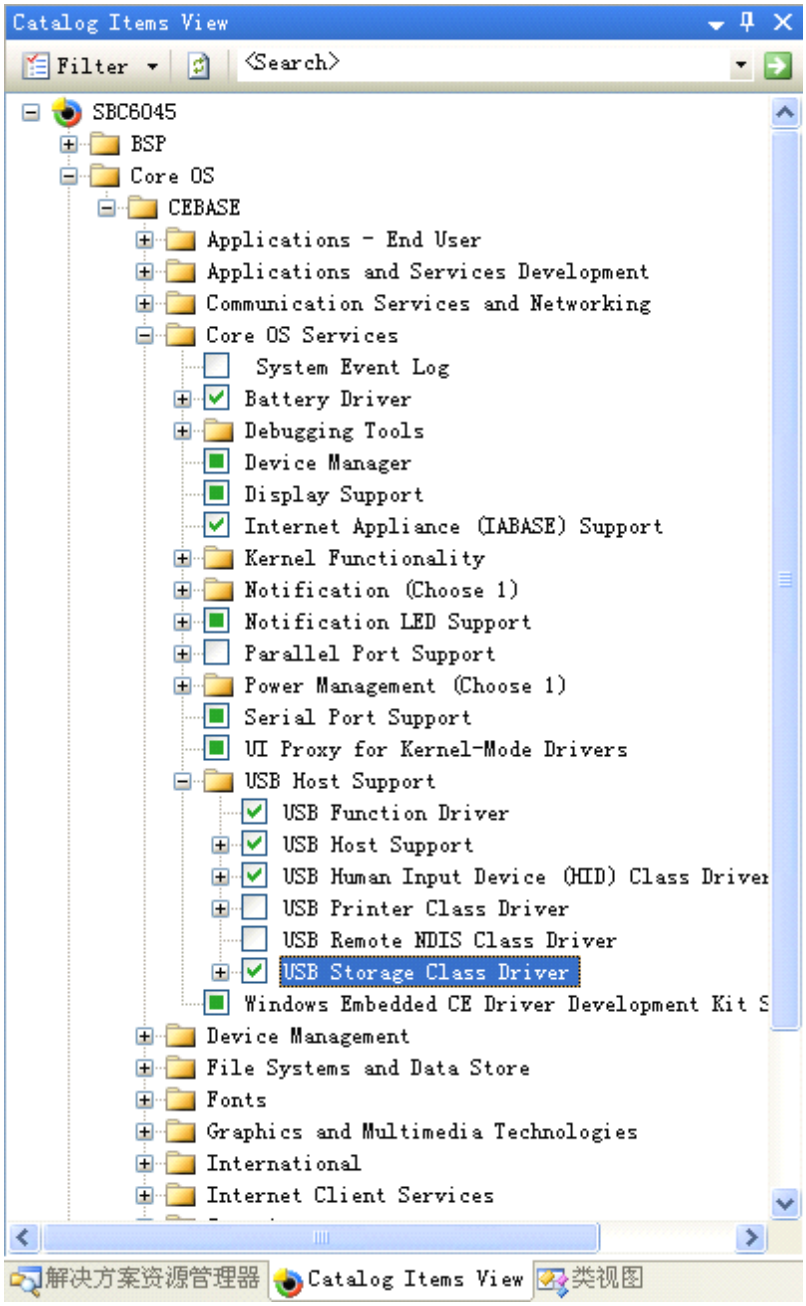



Figure 13.4.11

12. Add the following options following the steps of add USB relevant modules as described above:

| |
|---|
| Core OS->CEBASE ->Communication Services and Networking->Networking-General->Domain Discovery |

Core OS->CEBASE ->Communication Services and Networking->Networking-General ->Extended DNS Querying and Update(DNSAPI)

Here we remove the existing default option below:

Core OS->CEBASE -> Communication Services and Networking->Networking-General-> TCP/IPv6 Support

Continue to add the following options:

Core OS->CEBASE->File Systems and Data Store->Storage Manager->Storage Manager Control Panel Applet

Core OS->CEBASE->International->Locale Specific Support-> Chinese （Simplified） ->Fonts->SimSun&NsimSun(Choose 1)->SimSun&NsimSun

Core OS->CEBASE->International->Locale Specific Support-> Chinese （Simplified） ->GB1803030 Data Converter-

Core OS->CEBASE->International->Locale Specific Support-> Chinese （Simplified）-> Monotype Imaging AC3 Font Compression

Device Drivers->SDIO->SDIO Host->SDIO Standard Host Controller

Device Drivers->SDIO->SDIO Memory->SD Memory

Device Drivers->USB Function->USB Function Clients->Mass Storage

Device Drivers->USB Function->USB Function Clients->serial

13. Set compilation type as SBC6045 ARMV4I Release, see Figure 13.4.12:

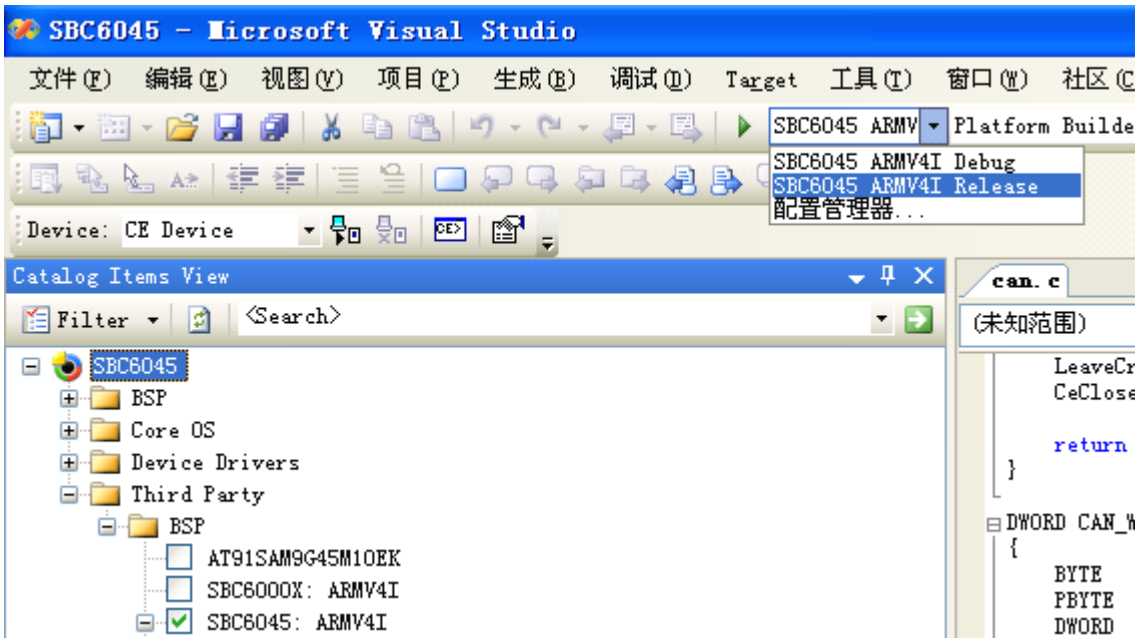



Figure 13.4.12

14. Click Project->SBC6045 properties in VS2005 , see Figure 13.4.13:

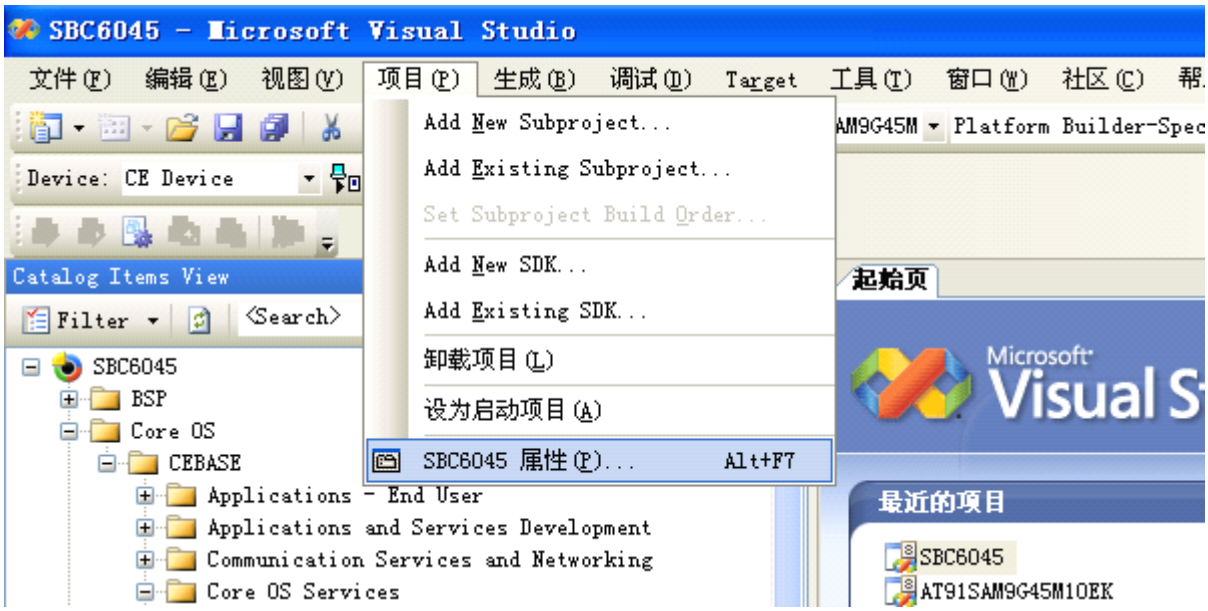



Figure 13.4.13

15. In “Locale” Option,select the language you need, than tick Enable eboot space in memory(IMAGE=1) in Build Options, as shown in Figure 13.4.15, click OK to finish the settings of compilation.

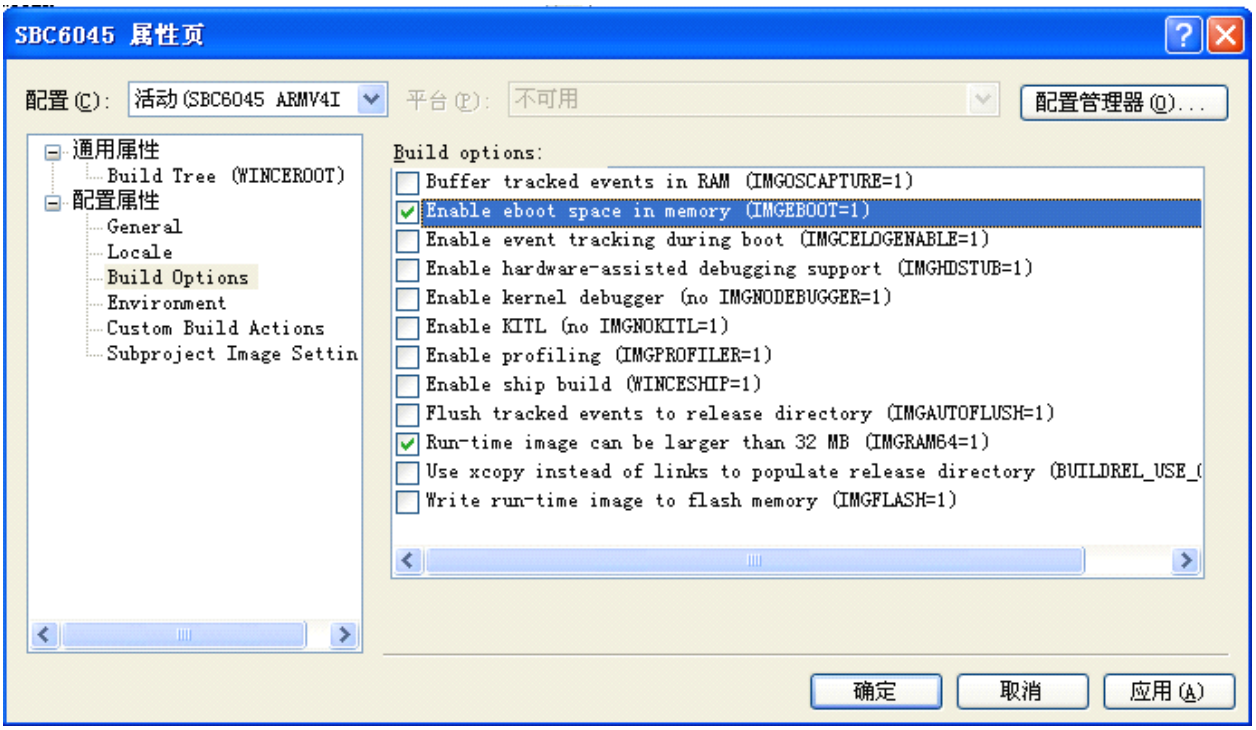



Figure13.4.15

16. As shown in Figure 13.4.16, tick the driver of the module you need SBC6045 board, in View->Other window->Catalog Items View->SBC6045->Third Party->SBC6045: ARMV4I in VS2005. (For details of each driver module please refer to Section 13.5) .

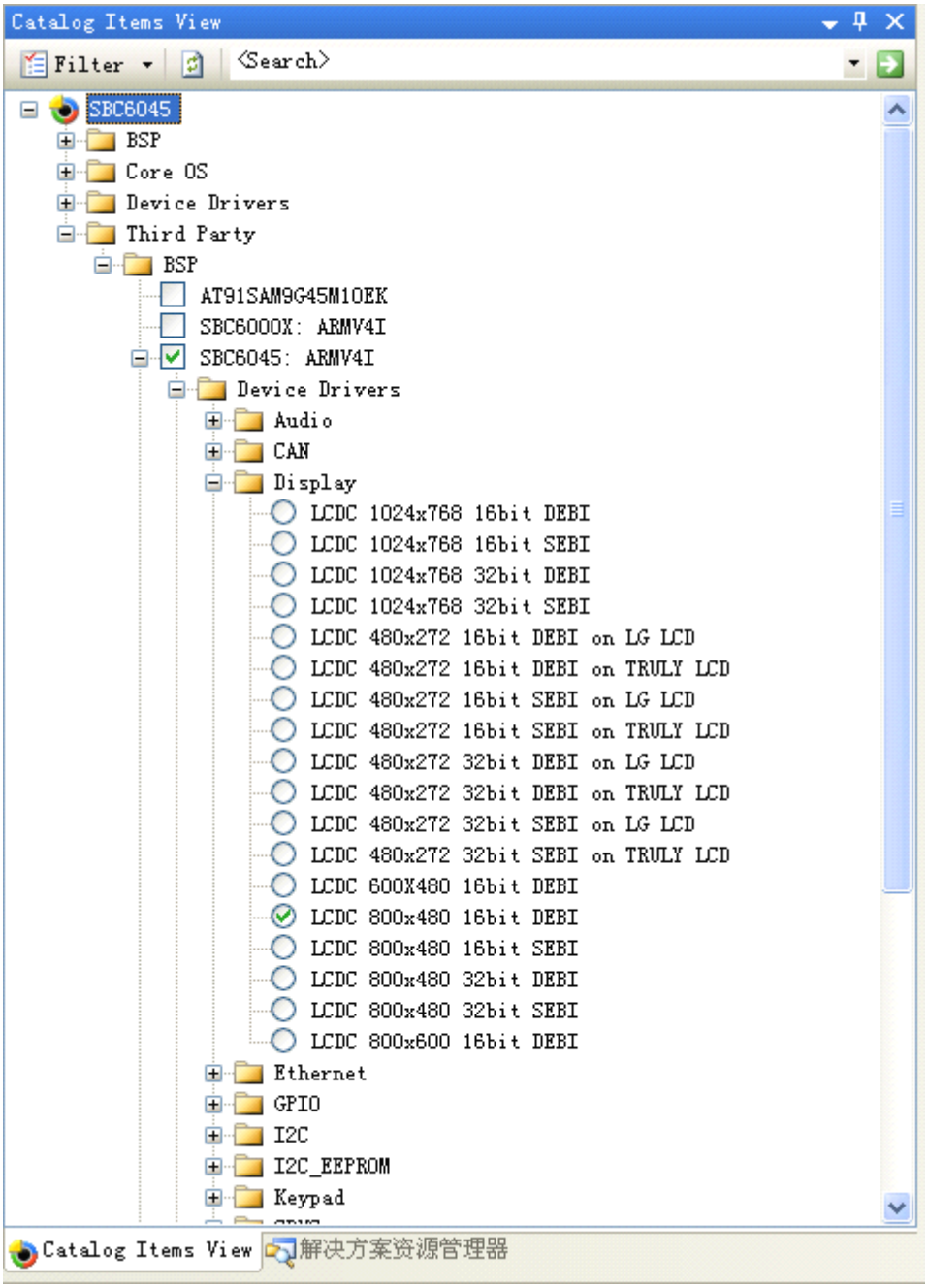



Figure 13.4.16

17. As shown in Figure 13.4.17, click Build->Advanced Build Commands->Sysgen of VS2005 to start compiling:

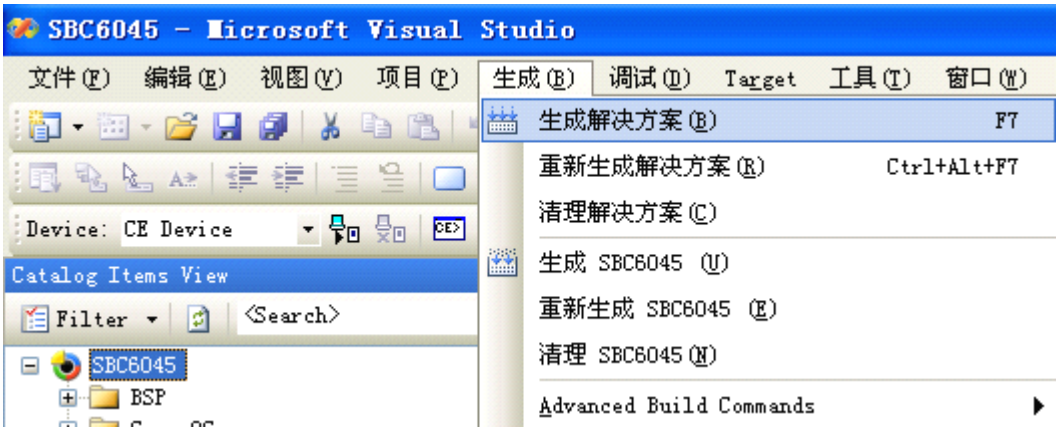



Figure13.4.17

18. As the compilation may take a long time (depends on the hardware configurations of development workstation), please wait. After the system compilation is successful, VS2005 will export the following information as shown in Figure 13.4.18:

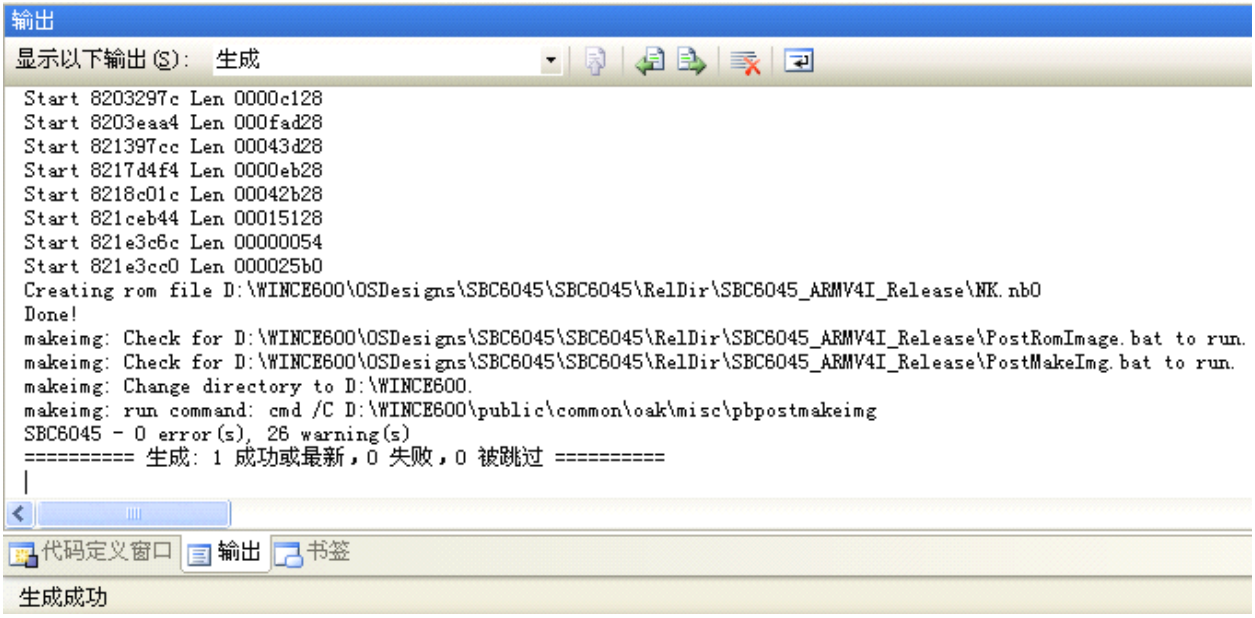


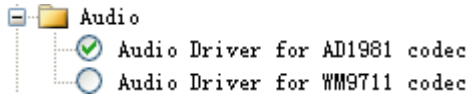
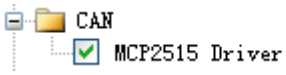
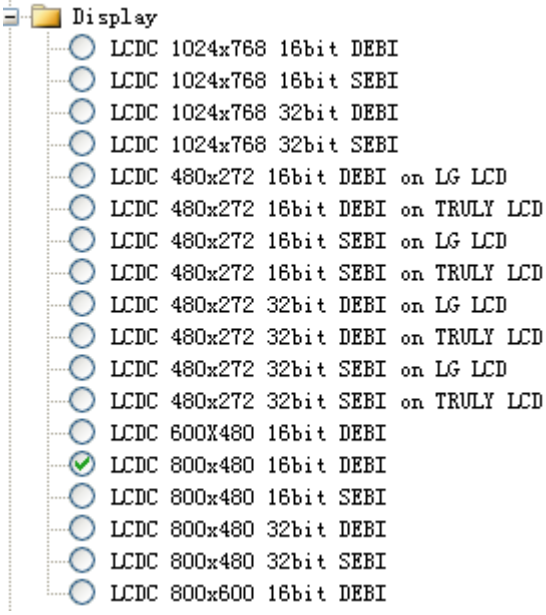
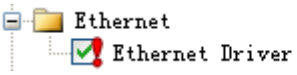
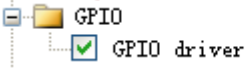

Figure 13.4.18

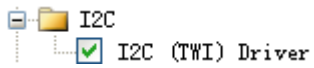
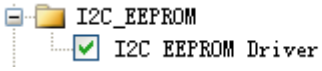
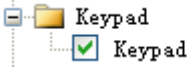
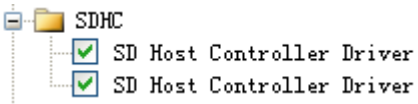
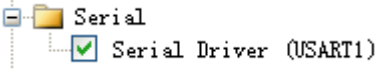
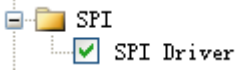
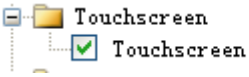
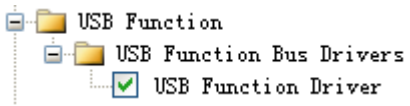
19. The moment WinCE 6.0 image files, i.e.: STEPLDR.nb0, EBOOT.nb0, NK.nb0, STEPLDR.bin, EBOOT.bin, NK.bin, will be generated under D:\WINCE600\OSDesigns\SBC6045\ RelDir\ SBC6045_ARMV4I_Release folder.

## 13.5 Modules of SBC6045 WinCE 6.0 BSP driver

After SBC6045 WinCE 6.0 BSP is installed and the project based on the BSP is initially customized following the instructions of Section 13.4, click View->Other window->Catalog Items View->SBC6045->Third Party-> SBC6045: ARMV4I in VS2005, you will see the modules of each driver module of SBC6045 WinCE 6.0 BSP, you can choose them on your demand. The table below will describe the functions of each module in details.

| Module | Functions | Remarks |
|---|---|---|

| | | |
|---|---|---|
|  | Audio Driver module | If you need use the play function of external audio chip AD1981 chip on SBC6045 board, directly tick this module in system customization. |
|  | The driver of CAN Bus on SBC6045 | If you need use CAN Bus. directly tick this module and SPI module in system customization. |
|  | The driver of TFTLCD on SBC6045 | TFT LCD support should add this module in system .And customer should select the correspond type of LCD the system have.<br>✧ 4.3 inch select ”LCDC 480x272 16bit DEBI on TRULY LCD”<br>✧ 5.6 inch select “LCDC 600X480 16bit DEBI”<br>✧ 7 inch select ”LCDC 800x480 16bit DEBI”<br>✧ 10.4 inch select ”LCDC 800x600 16bit DEBI” |
|  | AT91SAM9G45 Ehernet driver | If you need use Ethernet. directly tick this module in system customization. |
|  | AT91SAM9G45 Gerneral IO driver | If you need this function. directly tick this module in system customization. |

| | | |
|---|---|---|
|  | The driver of I2C interface on SBC6045 board. | If you need this function. directly tick this module in system customization. |
|  | External Eeprom driver of SBC6045 | If you need use EEPROM in systerm. directly tick this module and I2C interface driver in system customization. |
|  | The driver of IO key interface on SBC6045 board outgoing from AT91SAM9G45 | If you need this function. directly tick this module in system customization. |
|  | AT91SAM9G45 SD card controller driver | If you need SD function. directly tick this module in system customization. |
|  | Driver of USART serial port on SBC6045 outgoing from AT91SAM9G45 chip | If you need use USART on SBC6045. directly tick this module in system customization. |
|  | The driver of SPI interface on SBC6045 board. | If you need this function. directly tick this module in system customization. |
|  | SBC6045 touch screen driver | Touch screen function support directly tick this module in system customization. |
|  | USB Device driver on SBC6045 | SBC6045 USB Device support . directly tick this module in system customization. |

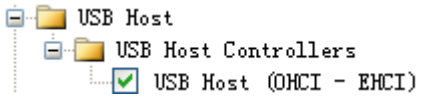
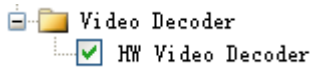
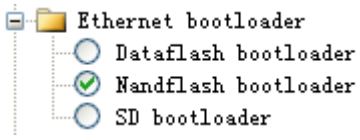
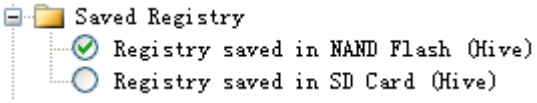
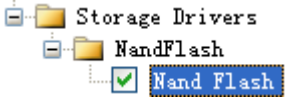
| | | |
|---|---|---|
|  | USB Host driver on SBC6045 | SBC6045 USB Host support, directly tick this module in system customization. |
|  | AT91SAM9G45 on chip hardware video decoder | Need hardware video decoder function directly tick this module in system customization. |
|  | SBC6045 boot mode select | Boot mode select |
|  | Hive Registry | If you select Hive Registry, this module should be added when you customize the system. |
|  | NAND flash driver | To use the rest NAND flash partition as storage after storing WinCE system, add this module when customizing the system. |

Figure 13.5.1

## 13.6 How to enter EBOOT menu

1. Connect Debug serial port J27 of SBC6045 main board to serial port of development workstation using the serial port cable supplied with SBC6045 main board, open CD\ 03 WinCE 6.0 Kit\04 Tools\dnw.exe see Figure 13.6.1.


Figure 13.6.1

2. Click DNW Configuration->Options, select baudrate as 115200 (see Figure 13.6.2):

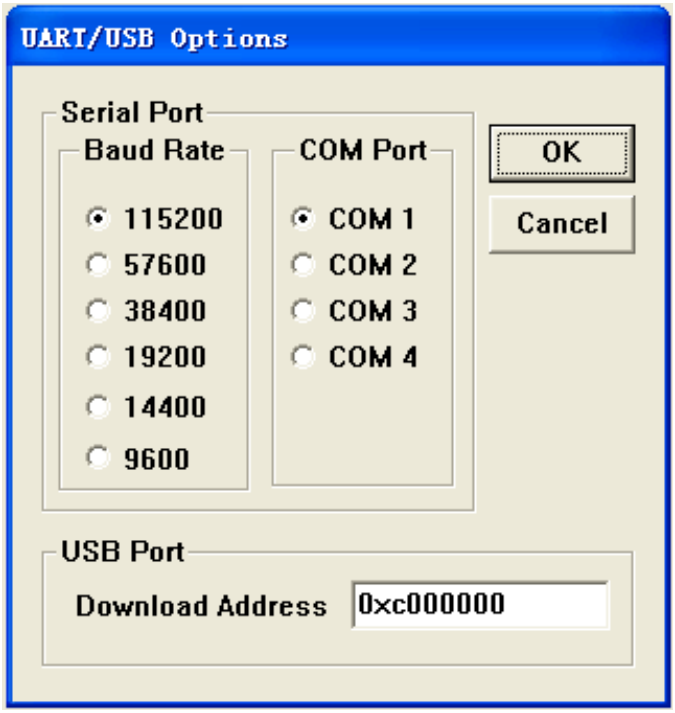



Figure13.6.2

3. Open serial port of development workstation (As shown in Figure 13.6.3, click DNW Serial Port-> Connect).

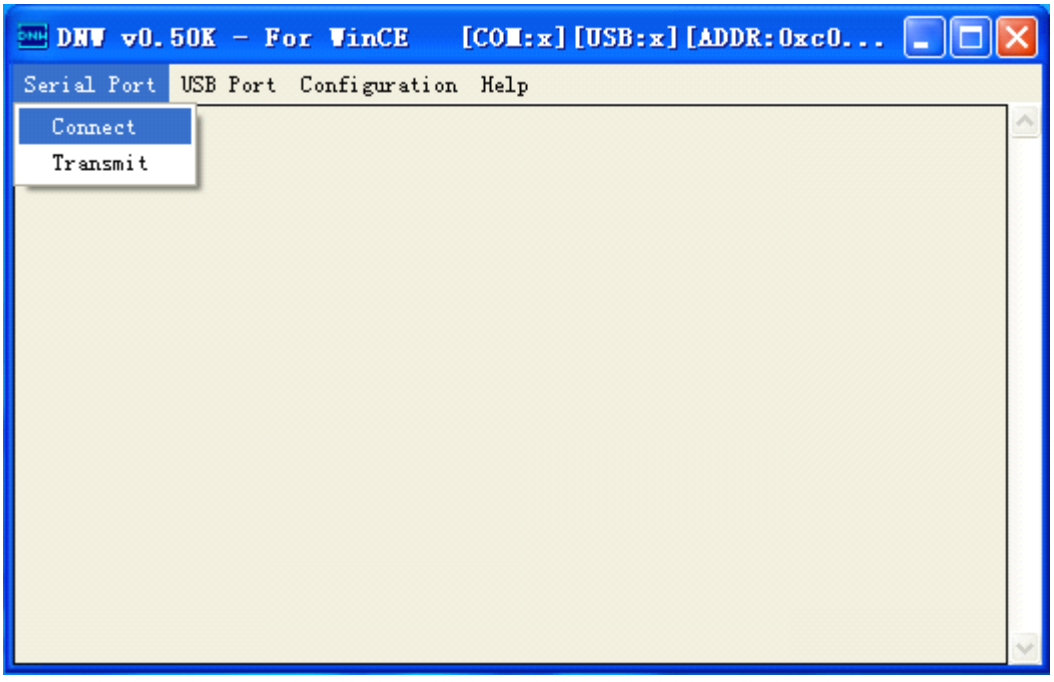



Figure13.6.3

4. Power on SBC6045 main board, SBC6045 serial port print information display enter countdown stage (the default countdown time is 5S after power-on of SBC6045 main board). Press SPACE key on DNW interface of development workstation, the moment SBC6045 EBOOT serial port output information as shown in Figure13.6.4 will appear:

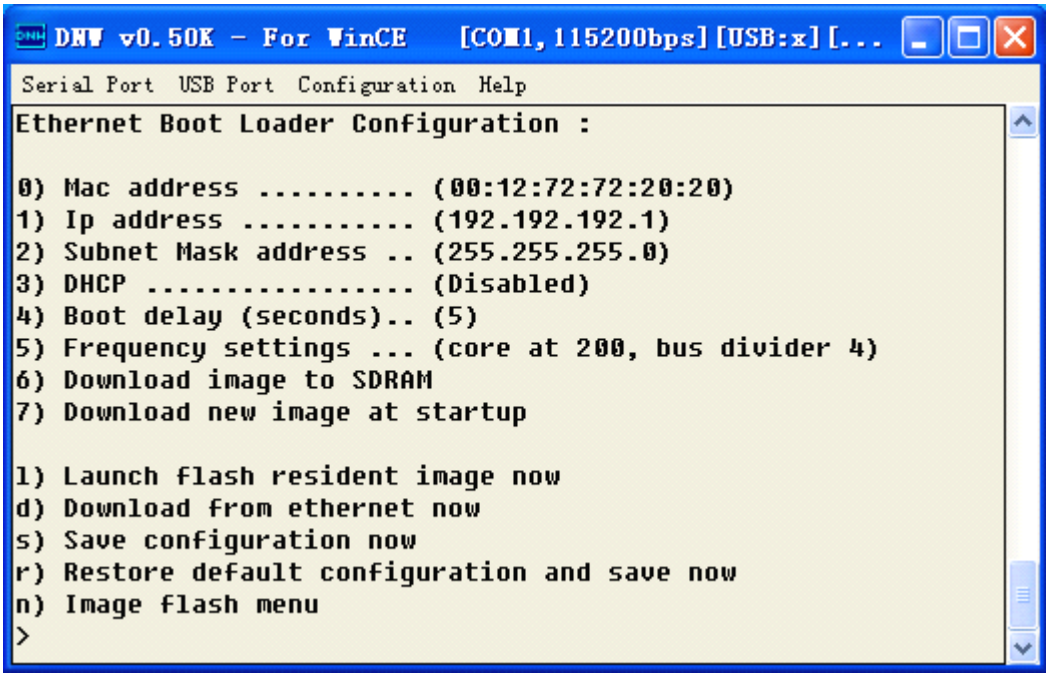



Figure 13.6.4

## 13.7 EBOOT menu commands

Table13.7.1 EBOOT menu commands

| Command | Items | Functions |
|---|---|---|
| 0 | Mac address ..... (00:12:72:72:20:20) | To set SBC6045 MAC Address |
| 1 | Ip address ..... (192.192.192.1) | To set SBC6045 IP address |
| 2 | Subnet Mask address .. (255.255.255.0) | To set SBC6045 subnet mask. |
| 3 | DHCP ..... (Disabled)<br>DHCP ..... (Enabled) | To set whether to dynamically allocate IP address for SBC6045 |
| 4 | Boot delay (seconds).. (5) | To set the delay for EBOOT to launch NK |
| 5 | Frequency settings ... (core at 200, bus divider 2) | To set master frequency and bus frequency of AT91SAM9G45 chip |
| 6 | Download image to Flash<br>Download image to SDRAM | Download NK to SDRAM or NAND flash |
| 7 | Launch existing flash resident image at startup<br>Download new image at startup | EBOOT loads existing flash resident image or downloads NK again at startup |

| | | | |
|---|---|---|---|
| l | | Launch flash resident image now | Load NK |
| d | | Download from ethernet now | Download NK over Ethernet |
| s | | Save configuration now | Save current EBOOT settings in flash partition |
| r | | Restore default configuration and save now | Resume default configurations of EBOOT. |
| n | 1 | Erase all sectors | Erase all sectors of NAND flash |
| | 2 | Enter manually the image parameters | Manually enter booting parameters |
| | 3 | Quit... | Exit n sub-menu |

1. Mac address ..... (00:12:72:72:20:20) （command 0）

**Function:** set SBC6245 net card MAC address

**Operating instructions:** Enter “0” in EBOOT menu in DNW interface, the following information will appear:

```
Enter new Mac Address:
```

Enter MAC address of SBC6245 main board net card in DNW interface and press ENTER key, here we set it as 00:11:22:33:44:55:

```
Enter new Mac Address: 00:11:22:33:44:55
```

2. Ip address ..... (192.192.192.1) （command1）

**Functions:** set SBC6245 main board net card IP address in EBOOT menu in DNW interface.

**Operating instructions:** Enter “1” in EBOOT menu in DNW interface, the following information will appear:

```
Enter new IP address:
```

Enter IP address of SBC6245 main board net card in DNW interface and press ENTER key, here we set it as 192.192.192.1:

```
Enter new IP address: 192.192.192.1
```

3. Subnet Mask address .. (255.255.255.0) （Command 2）

**Functions:** set SBC6245 net card subnet mask

**Operating instructions:** Enter “2” in EBOOT menu in DNW interface, the following prompt information will appear:

```
Enter new subnet mask:
```

Enter subnet mask address of SBC6245 net card in DNW interface and press ENTER key, here we set it as 255.255.255.0:

```
Enter new subnet mask: 255.255.255.0
```

4. DHCP ..... (Disabled) （command3）

**Functions:** whether to allocate dynamic IP for SBC6245. if this function is enabled, the IP address of SBC6245 will be allocated by the online DHCP server of SBC6245; if this function is disabled. The IP address of SBC6245’s IP is the IP address that set by command 1.

**Operating instructions:** enter “3” in EBOOT menu, the 3rd item will shift between

| 3) DHCP ..... (Enabled) |
|---|

and

| 3) DHCP ..... (Disabled) |
|---|

they mean Enable or Disable DHCP function respectively.

5. Boot delay (seconds).. (5) （command 4）

**Functions:** set the time delay of SBC6245 in system auto booting, if EBOOT waits for a time longer than that time, the system will be automatically booted. If you press SPACE in EBOOT menu in DNW interface within the set delay, EBOOT will enter EBOOT menu.

**Operating instructions:** Enter “4” in EBOOT menu in DNW interface, then enter preset time T ( 0<T<256), and press Enter key, the modifications of configurations must be saved by executing **,** and will not work unless you have restarted the system.

*Notes: the preset time shall not be 0, otherwise the system will automatically enter Windows CE system after restarting, in this case, user can not enter EBOOT menu by pressing space key.*

6. Frequency settings ... (core at 200, bus divider 2) （command 5）

**Functions:** set master frequency and bus frequency of SBC6245 CPU AT91SAM9G45

**Operating instructions:** Enter “5” , DNW in EBOOT menu in DNW interface, the following information will appear:

| Enter new Core Frequency (old frequency is 200 MHz) |
|---|

Enter master frequency of CPU in DNW, here we set it as 200MHz, enter “200” in DNW, then press ENTER

| Enter new Core Frequency (old frequency is 200 MHz) 200 |
|---|

The moment the following information will appear in DNW:

| Enter new Bus divider (old divider is 2) |
|---|

Enter “2” in DNW, it means bus divides CPU frequency into two sub-frequency, that is to say, busfrequency is 100MHz, enter 2 in DNW and press ENTER

| Enter new Bus divider (old divider is 2) 2 |
|---|

7. Download image to Flash （command 6）

**Functions:** command 6 decides whether the NK.bin file that EBOOT downloads from development workstation will finally be burnt into NAND flash.

**Operating instructions:** enter “6” in EBOOT menu, the 6 th item will shift between

| 6) Download image to SDRAM |
|---|

and

| 6) Download image to Flash |
|---|

If it shifts to “Download image to Flash”, the NK.bin file that EBOOT downloads from development workstation will finally be burnt into NAND flash, and will launch SDRAM and run the system; if it shifts to “Download image to SDRAM”, the NK.bin file that EBOOT downloads from development workstation will not be burnt into NAND flash, directly launch it in SDRAM and run the system.

8. Launch existing flash resident image at startup （command 7）

**Functions:** Command 7 decides whether EBOOT directly launch NK of NAND flash or download

it and burn NK again at startup.

**Operating instructions:** enter “7” in EBOOT menu, the 7th item will shift between

```
7) Launch existing flash resident image at startup
```

and

```
7) Download new image at startup
```

If it shifts to “Download new image at startup”, EBOOT need download and burn NK again at startup. If it shifts to”Launch existing flash resident image at startup”, EBOOT will directly launch NK from NAND flash.

9. Launch flash resident image now （command l）

**Functions:** Directly launch NK from NAND flash

**Operating instructions:** Enter “l” in EBOOT menu, EBOOT will start to launch NK from NAND flash.

10. Download from ethernet now（command d）

**Functions:** Directly download NK from Ethernet.

**Operating instructions:** Enter “d” in EBOOT menu, EBOOT will start to download NK from Ethernet.

11. Save configuration now（command s）

**Functions:** save current EBOOT configurations in Dataflash or NAND flash.

**Operating instructions:**Enter “s” in EBOOT menu, EBOOT will start to save current configurations of EBOOT.

12. Restore default configuration and save now（command r）

**Functions:** resume default configurations of EBOOT.

**Operating instructions:** Enter “r” in EBOOT menu, EBOOT will start to resume default

13. configurations of EBOOT. Image flash menu (command n)

**Functions:** a NAND flash operating command. Enter “n” in EBOOT menu, the moment the following information, namely the sub-menu image flash menu will appear in DNW.

```
Image Flash Menu :
1) Erase all sectors
2) Enter manually the image parameters
3) Quit...
```

- 1) Erase all sectors erase the whole NAND flash。
- 2) Enter manually the image parameters enter relevant information of NK image.
- 3) Quit... exit tehe sub-menu Image flash menu.

**Operating instructions:** Enter “n” in EBOOT menu, the moment EBOOT will export the sub-menu

image flash menu. Enter “1” in the sub-menu image flash menu, the system will erase all sectors of NAND flash; enter “2” in the sub-menu image flash menu, the moment the following information will appear in DNW

```
Enter the information found in the build report of the 'Make Image'. [entering 0 keeps present value]
Physical Start Address : (0x8006c000) : 0x
```

Here we enter the default value 0x8006c000 and press ENTER, the moment the following information will appear in DNW:

Enter the information found in the build report of the 'Make Image'. [entering 0 keeps present value]
Physical Start Address : (0x8006c000) : 0x8006c000
Starting ip : (0x8006d000) : 0x

We enter the default value 0x8006d000,the moment the following information will appear in DNW:

Enter the information found in the build report of the 'Make Image'. [entering 0 keeps present value]
Physical Start Address : (0x8006c000) : 0x8006c000
Starting ip : (0x8006d000) : 0x8006d000
Total ROM size : (0x2800000) : 0x

Then we enter default value 0x2800000 and press ENTER, the system will return to the sub-menu image flash menu. Enter 3 in the sub-menu image flash menu to exit the sub-menu image flash menu.

*Notes: Physical Start Address, Physical Start Address, Total ROM size, the three items related to NK*
*image files that we entered are the default information that EBOOT provides, they are very important*
*when EBOOT launch system, if they are incorrect, the system will not be launched normally. User can also execute the following steps to view the three parameters of NK.*

1. *After the compilation of SBC6045 project based on SBC6045 WinCE 6.0 is successful, click ->Open Release Directory in Build Window in VS2005. See Figure 13.7.1*

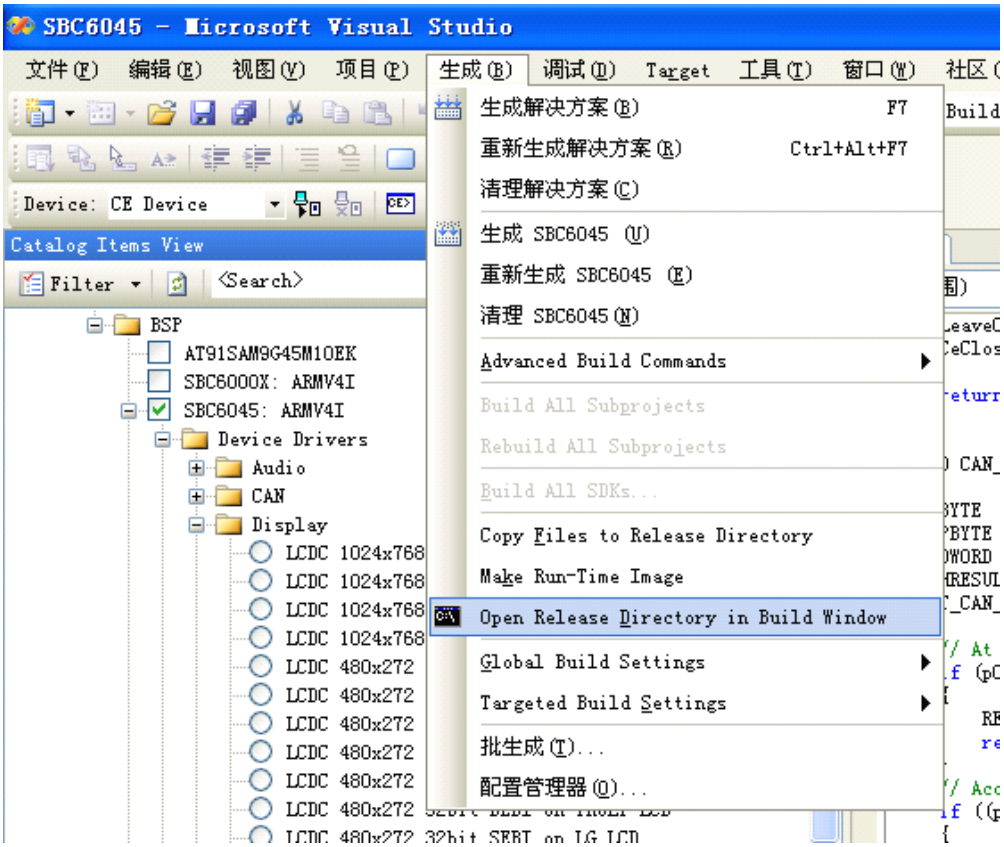



*Figure13.7.1*

*2. Enter viewbin nk.bin in the opened SBC6045 ARMV4I Release command window as shown in Figure 13.7.2, relevant information of NK will be displayed.*

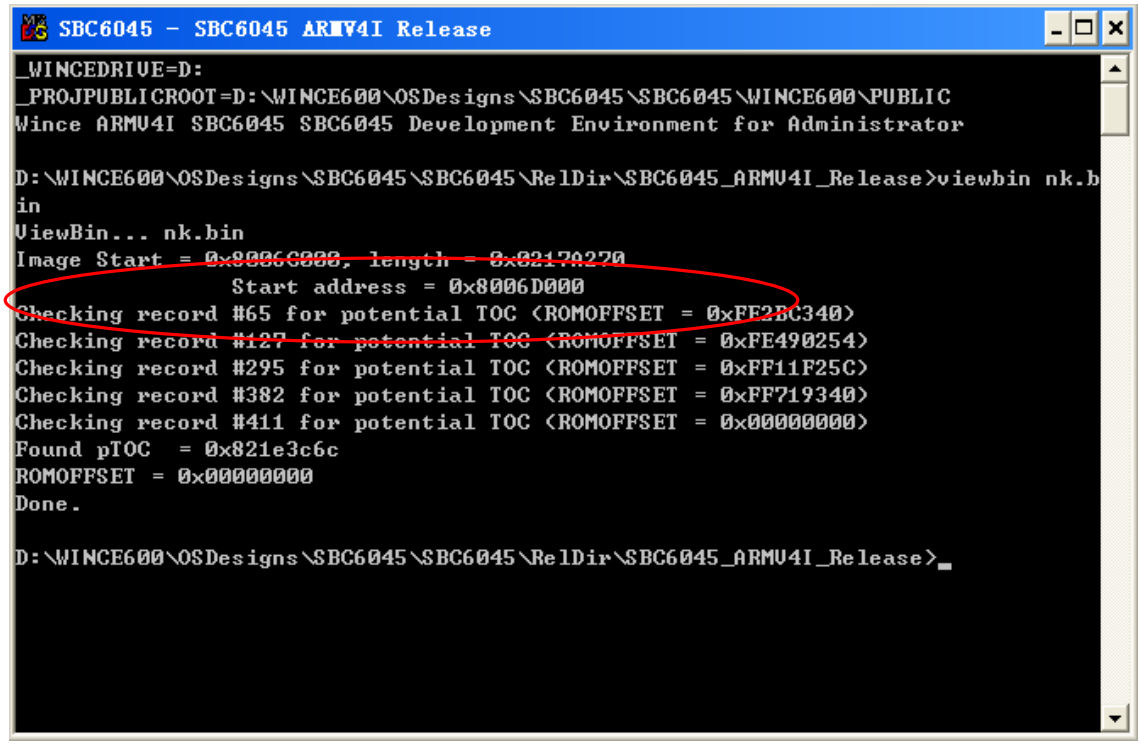



*Figure 13.7.2*

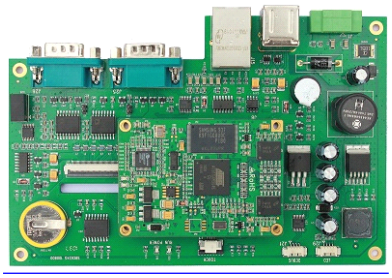
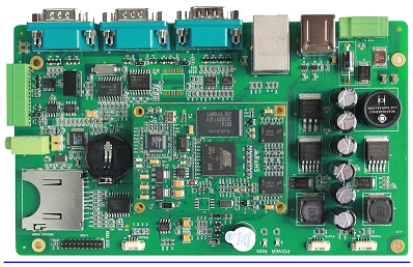
*Physical Start Address 即 Image Start*

*Starting ip 即 Starting address*

*Total ROM size 即 length*

*Please note that here we enter the maximum NK length 0x2800000 that EBOOT supports, say 40MB instead of specific 0x0217A270.*

# Similar products

| | |
|---|---|
|  |  |
| SBC6245 | SBC6345 |

# Technical support and Warranty Service

## Technical support service:

Embest provide one year free technical support service for all products Embest is selling. Technical support service covers:

- Embest embedded platform products software/hardware materials
- Assist customers compile and run the source code we offered
- Solve the problems accurate on embed software/hardware platform if user was following the instruction in the documentary we offered
- Judge if the product failure is existing

Special explanation, the situations listed below have NOT been included in the range of our free technical support service, Embest will handle the situation with discretion:

- Software/Hardware issues user met during the self-develop process
- Issues happened when users compile/run the embedded OS which was tailored by users themselves
- User's own application
- Problems happened during the modification of our software source code

## Maintenance service clause:

1. The products, which was used properly, will take the warranty since the day of the sale:

   PCB: Provide 12 months free maintenance service

2. The situations listed below have NOT been included in the range of our free maintenance service, Embest will charge the service fee with discretion:

   A. Can't provide valid Proof-of-Purchase, the identification label was tour up or illegible, the identification label was altered or not accord with the actual products;
   B. Didn'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 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 was 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 commitment which was committed by other institutions should be responsible by the institutions, Embest has nothing to do with that;

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

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

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

# Base 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.

# Value Added Services:

We will provide following value added services:

- Provided services of driver develop base on Embest embedded platform, like serial port, USB interface devices, LCD screen.

- Provided the services of control system transplant, BSP drivers develop, API software developing.
- Other value added services like power adapter, LCD parts.
- Other OEM/ODM services.
- Technically training.

# Contact Information

**Hotline**: +86-755-25635626-872/875/897
**Fax**: +86-755-25635626-666
**Pre-sales**: sales@embedinfo.com
**After-sales**: support@embedinfo.com
**Website**: 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)