41-S3-C2410X-102002

APPLICATION NOTE

S3C2410X
32-Bit RISC
Microprocessor
Revision 1

SAMSUNG ELECTRONICS

S3C2410X

32-BIT RISC MICROPROCESSOR APPLICATION NOTE

Revision 1

SAMSUNG ELECTRONICS

Important Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

Samsung reserves the right to make changes in its products or product specifications with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

This publication does not convey to a purchaser of semiconductor devices described herein any license under the patent rights of Samsung or others.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, for other applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use a Samsung product for any such unintended or unauthorized application, the Buyer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Samsung was negligent regarding the design or manufacture of said product.

S3C2410X RISC Microprocessor
Application Note, Revision 1
Publication Number: 41-S3-C2410X-102002

Samsung Electronics' microcontroller business has been awarded full ISO-14001 certification (BSI Certificate No. FM24653). All semiconductor products are designed and manufactured in accordance with the highest quality standards and objectives.

Samsung Electronics Co., Ltd.
San #24 Nongseo-Ri, Kiheung-Eup
Yongin-City, Kyunggi-Do, Korea
C.P.O. Box #37, Suwon 449-900

TEL: (82)-(331)-209-3176
FAX: (82)-(331)-209-3262
Home-Page URL: Http://www.samsungsemi.com

Printed in the Republic of Korea

Table of Contents

Chapter 1 About SMDK2410 Board

Table of Contents (Continued)

Chapter 2 Toolkit & Debugging

Table of Contents (Continued)

Chapter 3 Programming Flash Memories

Chapter 4 System Design

List of Figures

List of Figures (Continued)

List of Tables

1 ABOUT SMDK2410 BOARD

SYSTEM OVERVIEW

SMDK2410 (Samsung MCU Development Kit) for S3C2410X is a platform that is suitable for code development of SAMSUNG's S3C2410X 16/32-bit RISC microcontroller (ARM920T) for hand-held devices and general applications.

The S3C2410X consists of 16-/32-bit RISC (ARM920T) CPU core, separate 16KB instruction and 16KB data cache, MMU to handle virtual memory management, LCD controller (STN & TFT), NAND flash boot loader, System Manager (chip select logic and SDRAM controller), 3-ch UART, 4-ch DMA, 4-ch Timers with PWM, I/O ports, RTC, 8-ch 10-bit ADC and touch screen interface, IIC-BUS interface, IIS-BUS interface, USB host, USB device, SD host & multimedia card interface, 2-ch SPI and PLL for clock generation.

The SMDK2410 consists of S3C2410X, boot EEPROM (flash ROM), SDRAM, LCD interface, two serial communication ports, configuration switches, JTAG interface and status LEDs.

SMDK2410 OVERVIEW

The SMDK2410 (Samsung MCU Development Kit) shows the basic system-based hardware design which uses the S3C2410X. It can evaluate the basic operations of the S3C2410X and develop codes for it as well.

When the S3C2410X is contained in the SMDK2410, you can use an in-circuit emulator (MULTI-ICE/OPENice32-A900).

This allows you to test and debug a system design at the processor level. In addition, the S3C2410X with MULTI-ICE/OPENice32-A900capability can be debugged directly using the MULTI-ICE/OPENice32-A900 interface.

Figure 1-1 shows SMDK2410 function blocks.

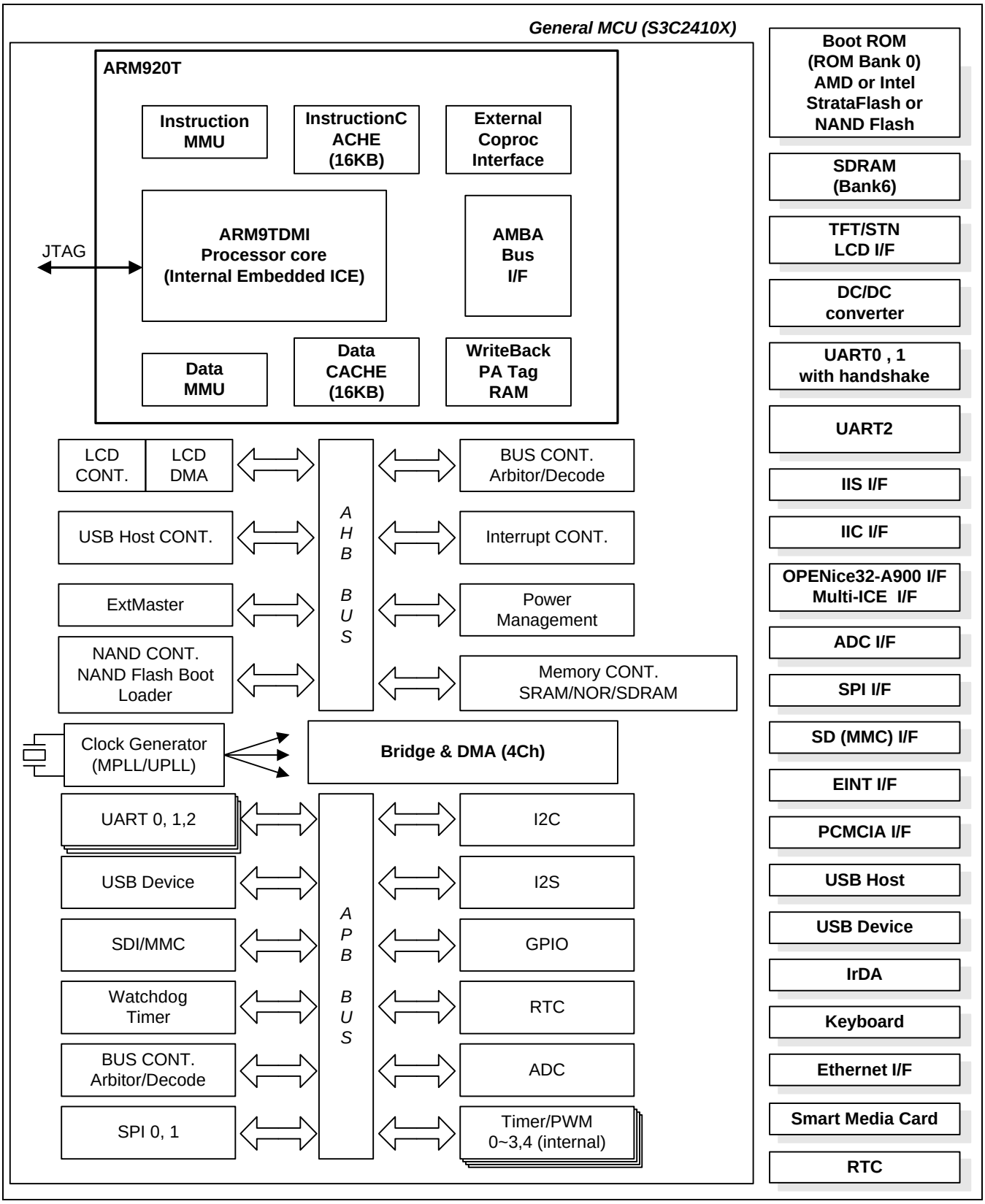


Figure 1-1. SMDK2410 Function Block Diagram

FEATURES

— S3C2410X: 16/32-bit RISC microcontroller

— X-tal operation or oscillator

— Boot ROM: AMD 8M bit 1EA (support halfword size boot ROM)
Intel StrataFlash 16M-byte x 3 (halfword: 16M-byte x 1 EA, word: 16M-byte x 2 EA): Unload (Option)
SAMSUNG NAND flash 64M-byte 1EA (smart media card).

— SDRAM: 64M-byte (32M-byte x 2)

— TFT/STN LCD and touch panel interface

— Three-channel UART (including IrDA)

— Two-port USB

— SD host (MMC) interface

— Smart media card

— JTAG port (MULTI-ICE/OPENice32-A900 interface)

— RTC X-tal input logic

— IIC with KS24C080

— ADC interface

— SPI Interface

— IIS interface (sound CODEC audio input/output)

— EINT interface

— IrDA interface

— 64 keyboard

— Ethernet interface

— PCMCIA interface

— Extension connector 34P * 3 EA

— LED display (debugging)

CIRCUIT DESCRIPTION

The SMDK2410 (Samsung MCU Development Kit) is designed to test S3C2410X and develop software while hardware is being developed. Figure 1-3 shows SMDK2410's block diagram.

POWER SUPPLY

SMDK2410 is operated by 1.8V for ARM core and 3.3V for I/O pad and several peripherals. SMDK2410 is supplied by ATX power which supports 3.3V, 5V and 12V and drives current at least 1.5A.

The SMDK2410 has distributed power plane, with power going separately to the MCU and the main power plane. For this reason, power jumpers including J21, J22, J23, J24, J25, J26, J27, J28, J29, J30, J31 and J32 are inserted.

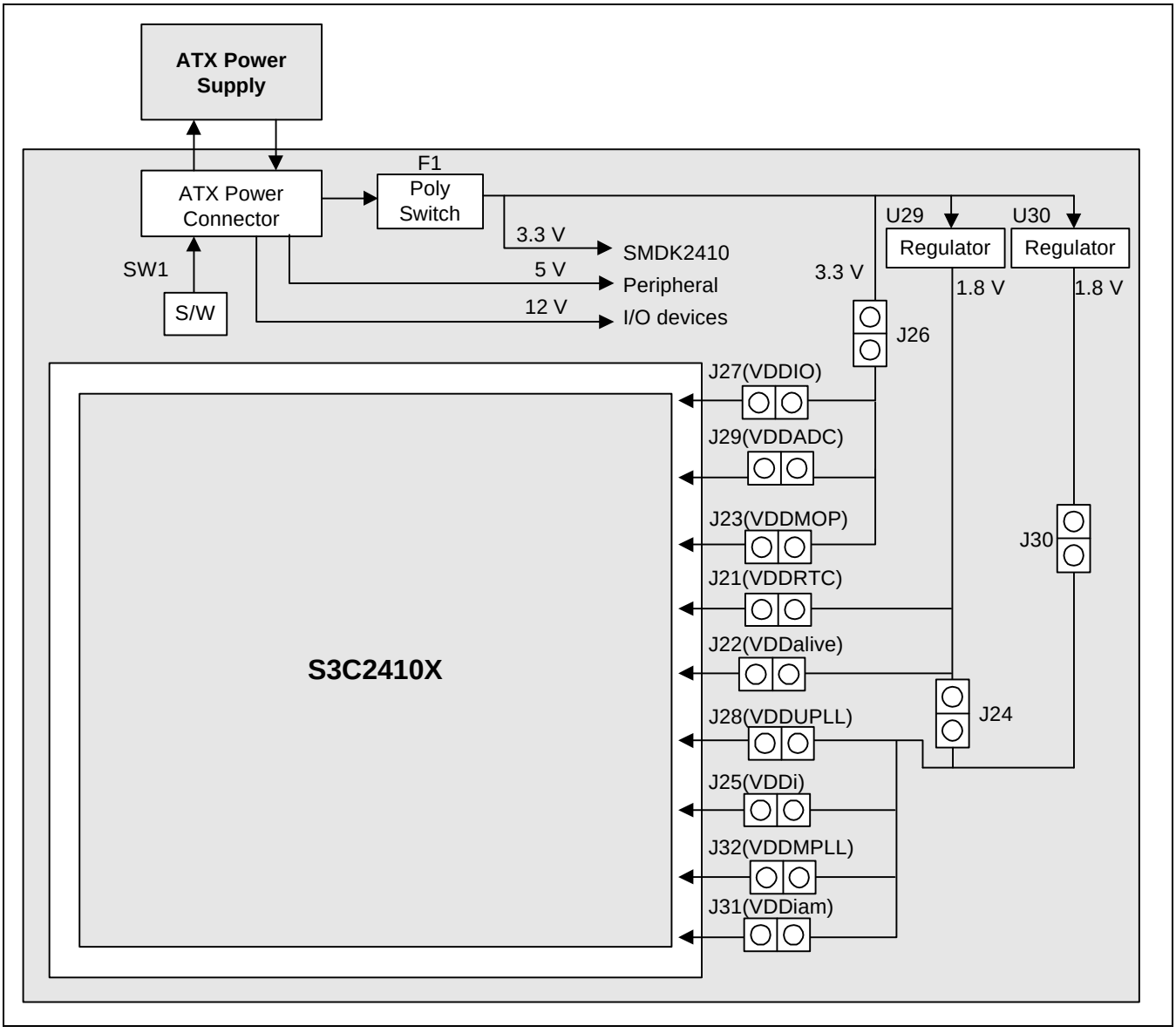


Figure 1-2. SMDK2410 Power Plane

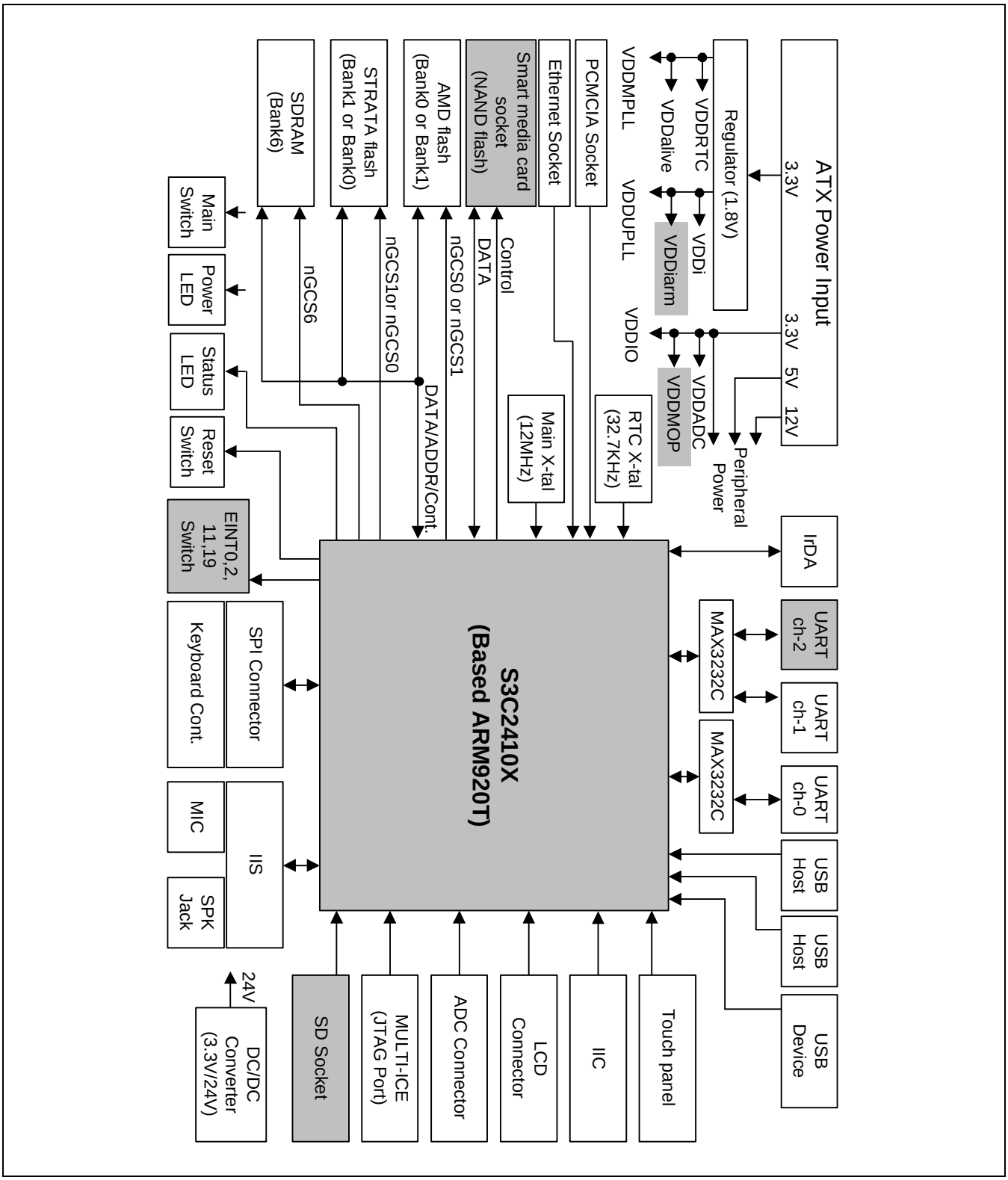


Figure 1-3. Detailed SMDK2410 Board Diagram

SMDK2410 SYSTEM CONFIGURATIONS

CLOCK SOURCE

EXTCLK or X-TAL can be selected for the system clock of S3C2410X and USB by the suitable setting of OM values.

Table 1-1. System Clock (MPLL) & USB Clock (UPLL)

PIN FUNCTIONS	OM[3:2]		DESCRIPTIONS
Clock source selection	0	0	MPLL: XTAL, UPLL: XTAL
	0	1	MPLL: XTAL UPLL: EXTCLK
	1	0	MPLL: EXTCLK, UPLL: XTAL
	1	1	MPLL: EXTCLK, UPLL: EXTCLK

RTC Clock

32.768KHz, X-tal is available in SMDK2410 as the RTC clock source.

NOTES:

1. Although the MPLL starts just after a reset, the MPLL output (Mpll) is not used as the system clock until the software writes valid settings to the MPLLCON register. Before this valid setting, the clock from external crystal or EXTCLK source will be used as the system clock directly. Even if the user wants to maintain the default value of the MPLLCON register, the user should write the same value into the MPLLCON register.
2. OM[3:2] is used to determine test mode when OM[1:0] is 11.

RESET LOGIC

The nRESET (system reset signal) must be held to low level at least 4 CLKs to recognize the reset signal and it takes 128 CLKs between the nRESET and internal nRESET. nRESET and nTRST (JTAG reset signal) are connected through 4.7K ohm resistor.

BOOT ROM (BANK0)

The data bus width of BANK0 can be configured in byte, half-word or word in S3C2410X. But byte data bus width is not supported in SMDK2410.

So, in the case of SMDK2410, halfword data bus width (AMD flash memory) and halfword or word data bus width (STRATA flash memory) access can be selected by the suitable jumper setting.

AMD flash or STRATA flash memory can be selected by using jumper (J6 & J9) option for boot ROM.
In the SMDK2410, the data bus width of AMD flash memory is fixed by halfword data width (16-bit) and STRATA flash memory can use halfword (16-bit) or word (32-bit).

But AMD flash and STRATA flash cannot be selected for BANK0 or BANK1 at the same time.
Data bus width of BANK0 should be set by memory type of BANK0. It is set by OM[1:0](J34 & J33).

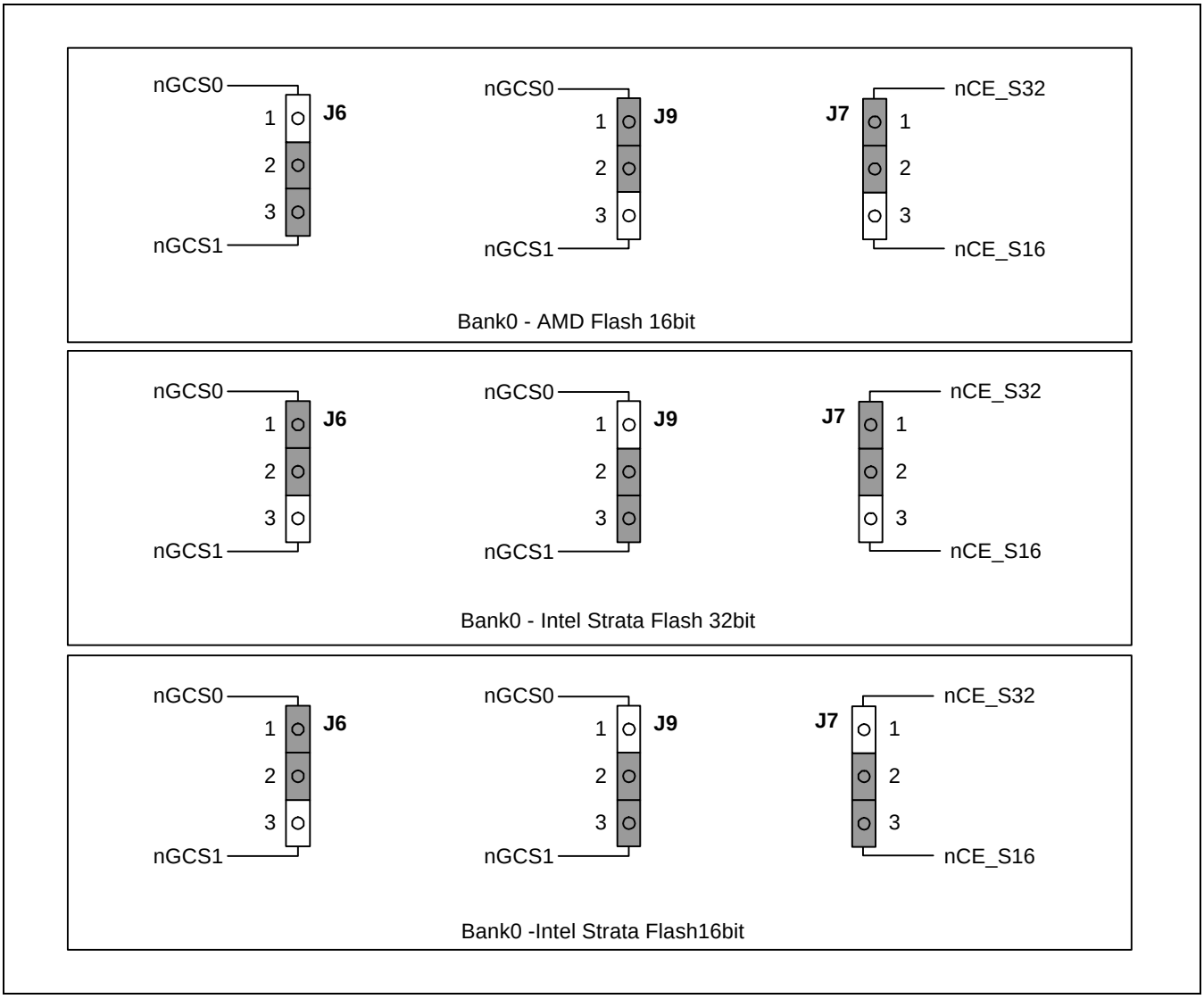


Figure 1-4. SMDK2410 Board Memory Configurations

Table 1-2. Memory Type and Data Bus Width

Pin Functions	J6		J9		J7		Descriptions
	(1-2)	(2-3)	(1-2)	(2-3)	(1-2)	(2-3)	
BANK0/BANK1 memory type selection and data bus width configuration	Open	Short	Short	Open	Short	Open	AMD flash memory : BANK0 STRATA flash memory : BANK1 Data bus width of BANK0 : Halfword Data bus width of BANK1 : Word
	Short	Open	Open	Short	Short	Open	AMD flash memory : BANK1 STRATA flash memory : BANK0 Data bus width of BANK0 : Word Data bus width of BANK1 : Halfword
	Short	Open	Open	Short	Open	Short	AMD flash memory : BANK1 STRATA flash memory : BANK0 Data bus width of BANK0 : Halfword Data bus width of BANK1 : Halfword

Table 1-3. Boot Memory Type and Bus width configuration OM[1:0]

Pin Functions	JP33 [OM0]	JP34[OM1]	Descriptions
Boot memory type and bus width configuration	Short	Short	NAND Boot
	Short	Open	Word (32-bit)
	Open	Short	Half Word (16-bit)
	Open	Open	Test Mode

NAND FLASH CONFIGURATION

To use a NAND flash memory, a user should set its type correctly using J3.

Table 1-4. NAND Flash Type Selection

Pin Functions	J3 [NCON]	Descriptions
NAND flash memory type selection	Short	512 Byte/Page, 3 Step Addressing
	Open	512 Byte/Page, 4 Step Addressing

Table 1-5. nWAIT and R/nB Signal Selection

S3C2410X	J2		Descriptions
	(1-2)	(2-3)	
Resvision 0	Short	Open	nWAIT
Resvision 1	Open	Short	R/nB (Ready / Busy output)

GENERAL I/O PORTS

The S3C2410X's general I/O ports are used for SMDK2410 key interrupt input and LED status display. The function of control switch and the status of LED can be defined by user software.

Table 1-6. General I/O Configurations on SMDK2410

General I/O Port Number	I/O Type	Descriptions
GPF[7:4]	Output	LED display
GPF0, GPF2, GPG3 & GPG11	Input	Key input pad (external interrupt input pins). (EINT0, 2, 11 & 19)

U4 (EPM7032) XDMA CHANNEL SELECTION

Table 1-7. U4 XDMA Channel Selection

Pin Functions	J4		J5		Descriptions
	(1-2)	(2-3)	(1-2)	(2-3)	
XDMA channel selection	Short	Open	Short	Open	nXDREQ0, nXDACK0
	Open	Short	Open	Short	nXDREQ1, nXDACK1

LCD INTERFACE

Both STN and TFT LCD controllers are equipped in the S3C2410X. STN/TFT LCD, touch panel and LCD inverter are supported in the SMDK2410.

NOTES:
To use SAMSUNG 240x320 TFT LCD, which is packed with SMDK2410, a user should set J20 to 3.3V.
1. Connect the SAMSUNG 240x320 TFT LCD connector to CON7 and CON9 on SMDK2410.
2. Short Pin2 and Pin3 of J20.

STN/TFT LCD Connector

The LCD video data signal (VD) and control signal are provided through three LCD connectors (CON5, CON7 and CON9) in the SMDK2410.
Voltage levels of VDD_LCDI, VD and control signals are supported by setting J20.

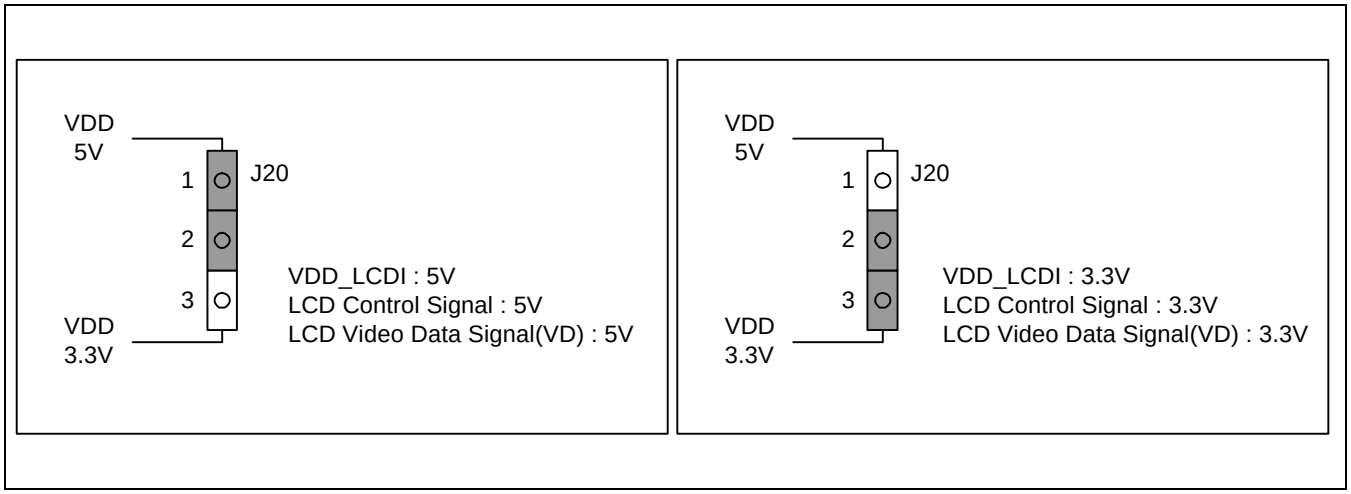


Figure 1-5. SMDK2410 Board LCD Configurations

Table 1-8. VDD_LCDI/VD/Control Signal Power Level

Pin Functions	J20		Descriptions
	(1-2)	(2-3)	
Power level selection	Short	Open	VDD_LCDI/LD/LCD control signal: 5V
	Open	Short	VDD_LCDI/LD/LCD control signal: 3.3V

Table 1-9. MONO/Gray STN LCD Connector (CON5) on SMDK2410

# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions
1	GND	**6**	LnDIS_OFF	**11**	LD0	**16**	LD5
2	VDD_LCDI	**7**	LVM	**12**	LD1	**17**	LD6
3	LVFRAME	**8**	–	**13**	LD2	**18**	LD7
4	LVLINE	**9**	–	**14**	LD3	**19**	–
5	LVCLK	**10**	–	**15**	LD4	**20**	VEE_LCD

Table 1-10. Color STN LCD film Connector (CON6) on SMDK2410

# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions
1	LVFRAME	**5**	VDD_LCDI	**9**	LD1	**13**	LD5
2	LVLINE	**6**	GND	**10**	LD2	**14**	LD6
3	LVCLK	**7**	VEE_LCD	**11**	LD3	**15**	LD7
4	LnDIS_OFF	**8**	LD0	**12**	LD4	**16**	GND

Table 1-11. TFT LCD Connector (CON7 40Pin and CON9 12Pin) on SMDK2410

# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions
1	LVCLK	**11**	GND	**21**	LD4	**31**	LLEND
2	LVLINE	**12**	LD10	**22**	LD5	**32**	–
3	LVFRAME	**13**	LD11	**23**	LD6	**33**	–
4	GND	**14**	LD12	**24**	LD7	**34**	VEE_LCD
5	GND	**15**	LD13	**25**	GND	**35**	–
6	LD19	**16**	LD14	**26**	LVM	**36**	–
7	LD20	**17**	LD15	**27**	VDD_LCDI	**37**	TSXP
8	LD21	**18**	GND	**28**	VDD_LCDI	**38**	TSXM
9	LD22	**19**	GND	**29**	VDD_LCDI	**39**	TSYP
10	LD23	**20**	LD3	**30**	GND	**40**	TSYM

# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions
1	LD0	**4**	LD8	**7**	LD16	**10**	LCDVF1
2	LLCD_ PWREN	**5**	LD2	**8**	LCDVF0	**11**	LD18
3	LD1	**6**	LD9	**9**	LD17	**12**	LCDVF2

TOUCH SCREEN

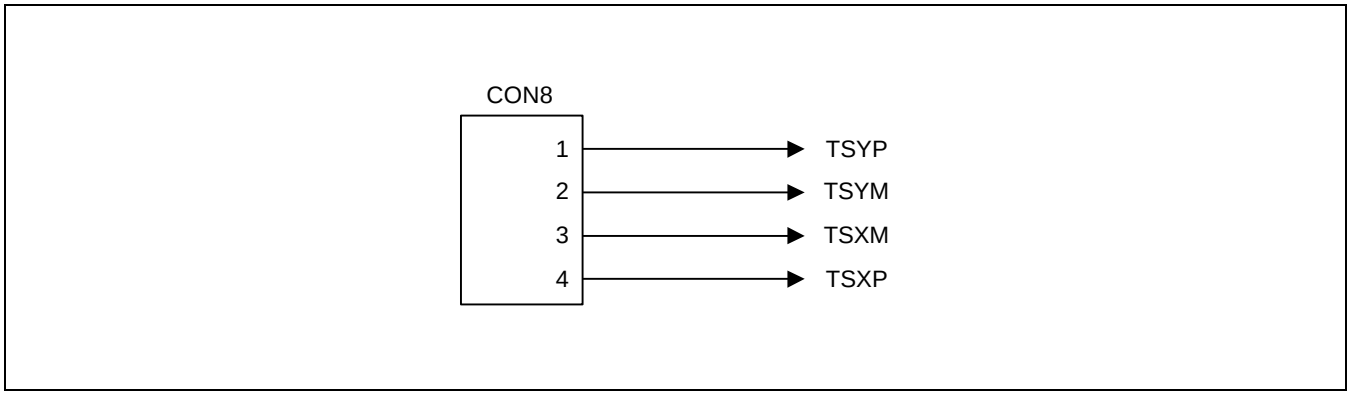


Figure 1-6. Touch Panel Film Connector on SMDK2410

KEYBOARD AND SPI Interface

Both S3C2410X and SMDK2410 support SPI. But, the S3C2410X does not provide keyboard.
So, the SMDK2410 has a keyboard with SPI (SPI ch1) and external keyboard controller (UR5HCSPI-SA).

Keyboard Connector

Keyboard is connected by the keyboard connector (CON2 & CON3). The SMDK2410 supports its function.

Table 1-12 Keyboard Configurations

Pin Functions	J12		J13		Descriptions
	(1-2)	(2-3)	(1-2)	(2-3)	
Keyboard Configuration	Short	Open	Open	Short	Enable Keyboard

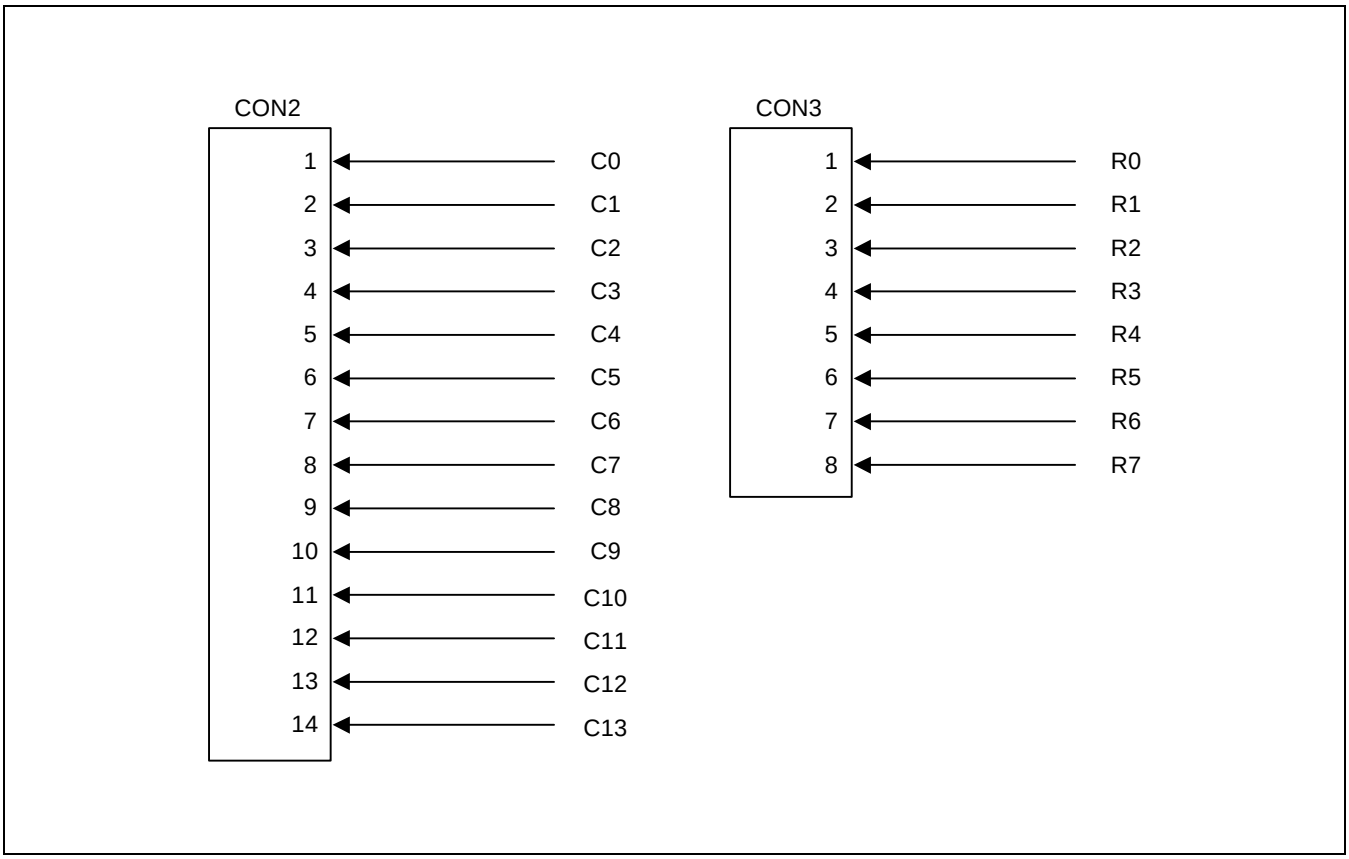


Figure 1-7. Keyboard Connector on SMDK2410

SPI Connector (SPI ch0)

Figure 1-8 shows the way SMDK2410 provides SPI (CON4) signals.

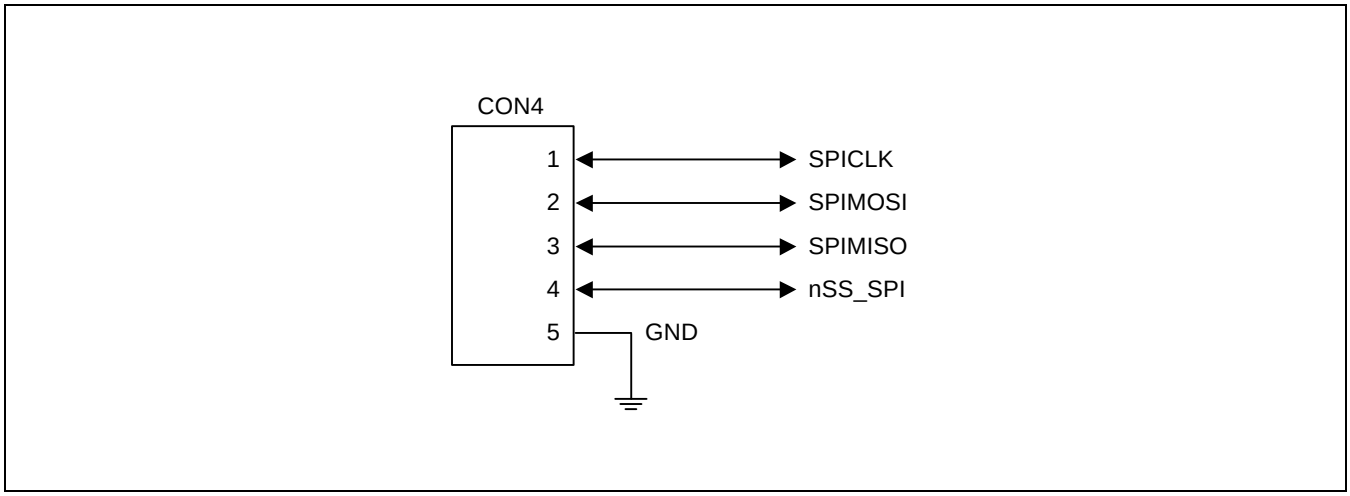


Figure 1-8. SPI Connector on SMDK2410

A/D CONVERTER INTERFACE

The S3C2410X has Analog to Digital Converter (ADC). The ADC has 8-ch analog input signals.
The SMDK2410 provides the ADC (CON18) signals as follows:

Table 1-13. ADC Interface on SMDK2410

# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions
1	VDD33V	**4**	AIN1	**7**	AIN4	**10**	AIN7
2	Avref	**5**	AIN2	**8**	AIN5	**11**	GND
3	AIN0	**6**	AIN3	**9**	AIN6	**12**	GND

SD HOST (MMC) INTERFACE

SD (MMC) is provided by the S3C2410X and SD card socket (CON11) is supported in the SMDK2410.

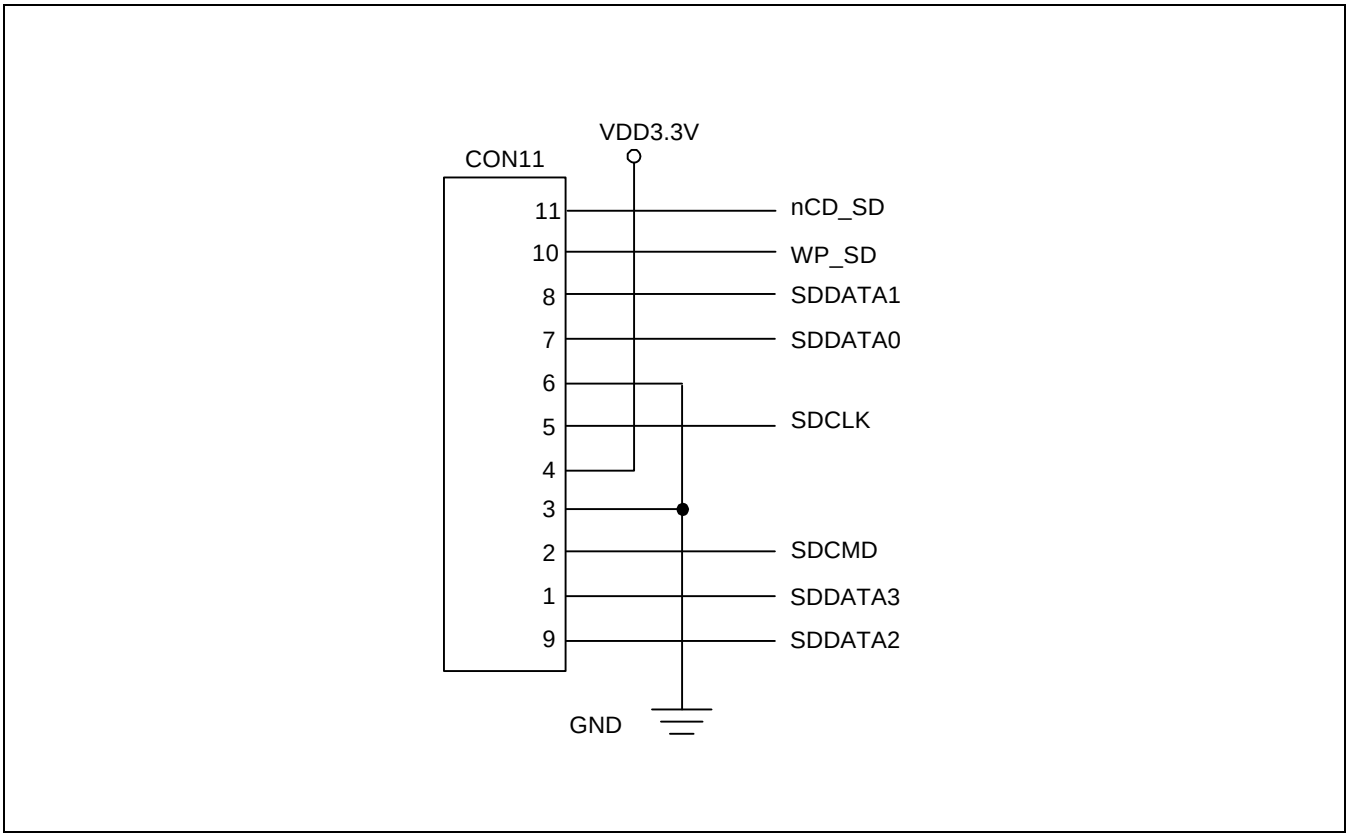


Figure 1-9. SD Card Socket on SMDK2410

IIC INTERFACE

Serial EEPROM S524C80D80 (KS24C080) access function is provided by SMDK2410 and there is no other interface or connector for IIC.

USB INTERFACE

Two USB ports A-type (CON14 & CON16) or One USB port A-type (CON14) and one USB port B-type (CON17) are supported by the SMDK2410.

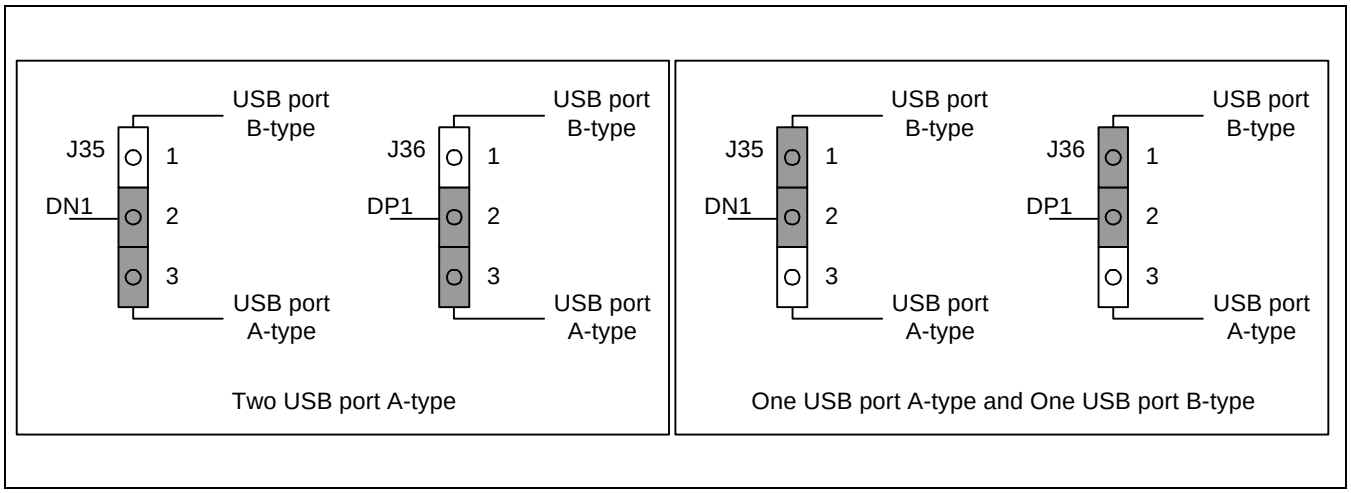


Figure 1-10. SMDK2410 Board USB Configurations

Table 1-14. USB Type Configurations

Pin Functions	J35		J36		Descriptions
	(1-2)	(2-3)	(1-2)	(2-3)	
USB type configurations	Open	Short	Open	Short	CON16 : USB Host (A-type)
	Short	Open	Short	Open	CON17 : USB Device (B-type)

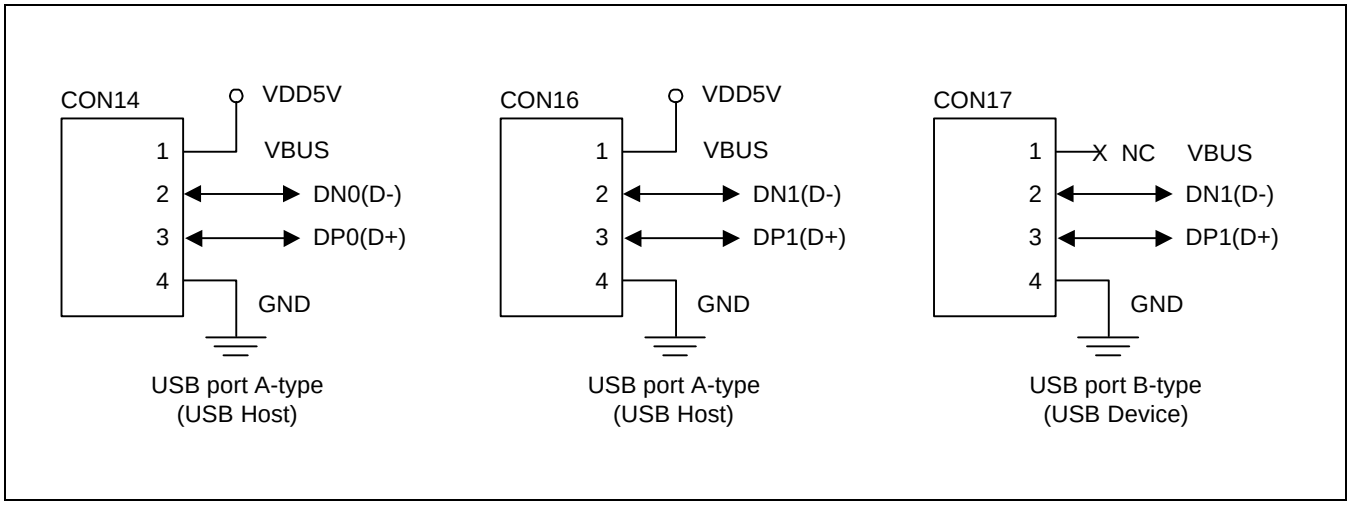


Figure 1-11. USB Ports on SMDK2410

UART INTERFACE

The S3C2410X UART unit provides three independent asynchronous serial I/O (SIO) ports including IrDA. In SMDK2410 board, a user can change the ports connected to connectors by setting related jumpers.

Table 1-15. UART Configurations

Pin Functions	J14, 15		J16, 17		J18, J19		J38	Descriptions
	(1-2)	(2-3)	(1-2)	(2-3)	(1-2)	(2-3)		
UART configurations	Open	Short	Short	Open	–	–	Open	P1: UART0, P2: UART1
	Short	Open	Open	Open	Open	Short	Open	P1: UART0, P2: UART2
	–	–	Open	Short	Open	Short	Short	P1: Modem

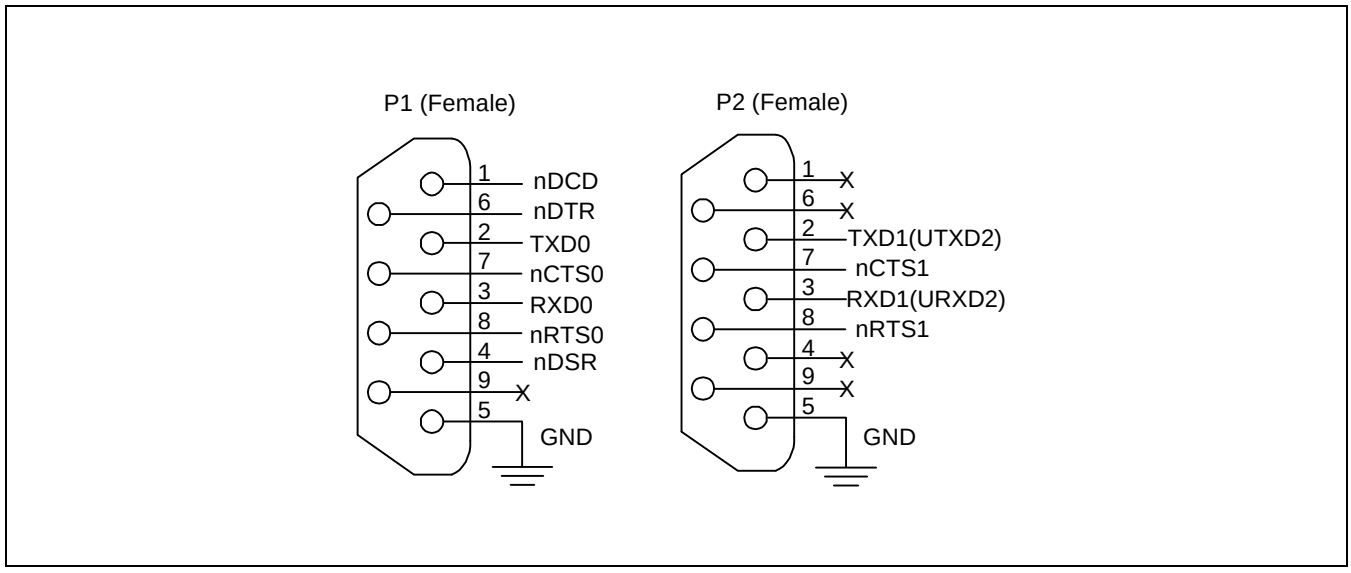


Figure 1-12. UART Ports on SMDK2410

IrDA INTERFACE

IrDA is supported by SMDK2410 and J19 and J18 should be set to UART2 (RXD2 and TXD2) for IrDA.

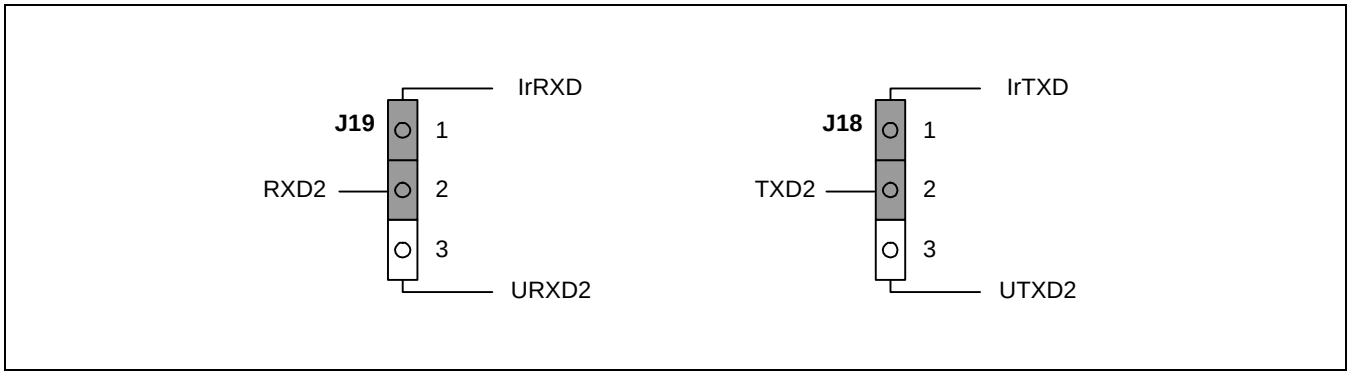


Figure 1-13. SMDK2410 Board IrDA Configurations

Table 1-16. IrDA Configurations

Pin Functions	J19		J18		Descriptions
	(1-2)	(2-3)	(1-2)	(2-3)	
IrDA configurations	Short	Open	Short	Open	Enable IrDA

OTHER JUMPERS

Some jumpers are reserved for special functions. So J1, J10,J11,J24, J37 and other holes named Jxx should be open.

LCD BOARD

Table 1-17. LCD Board Power Source Setting

Power Source	JP501		CON506 (Bottom Side)
	(1-2)	(2-3)	
ATX power 5V	Open	Short	ATX 4P connector connection
Main board VDD33V	Short	Open	Do not care

Before performing switch on the main board, connect the 4-pin connector of power supply to the LCD board.

EXTENSION CONNECTOR INTERFACE

Table 1-18. Extension Connector (CON12, CON13 & CON15) on SMDK2410

# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions
1	GND	**10**	DATA8	**19**	DATA17	**28**	DATA26
2	DATA0	**11**	DATA9	**20**	DATA18	**29**	DATA27
3	DATA1	**12**	DATA10	**21**	DATA19	**30**	DATA28
4	DATA2	**13**	DATA11	**22**	DATA20	**31**	DATA29
5	DATA3	**14**	DATA12	**23**	DATA21	**32**	DATA30
6	DATA4	**15**	DATA13	**24**	DATA22	**33**	DATA31
7	DATA5	**16**	DATA14	**25**	DATA23	**34**	–
8	DATA6	**17**	DATA15	**26**	DATA24	–	–
9	DATA7	**18**	DATA16	**27**	DATA25	–	–

# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions
1	GND	**10**	A8	**19**	A17	**28**	nWBE0
2	A0	**11**	A9	**20**	A18	**29**	nWBE1
3	A1	**12**	A10	**21**	A19	**30**	nWBE2
4	A2	**13**	A11	**22**	A20	**31**	nWBE3
5	A3	**14**	A12	**23**	A21	**32**	nWE
6	A4	**15**	A13	**24**	A22	**33**	nOE
7	A5	**16**	A14	**25**	A23	**34**	–
8	A6	**17**	A15	**26**	A24	–	–
9	A7	**18**	A16	**27**	nWAIT	–	–

# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions	# of pin	Descriptions
1	nGCS2	**10**	GPG7	**19**	nXDACK1	**28**	GND
2	nGCS1	**11**	GPG2	**20**	nXDREQ0	**29**	GND
3	nGCS4	**12**	GPG8	**21**	GPG5	**30**	nRESET
4	nGCS3	**13**	GPG3	**22**	nXDREQ1	**31**	VDD5V
5	nGCS7	**14**	GPG9	**23**	VDD18V	**32**	VDD33V
6	nGCS5	**15**	GPG4	**24**	GPG12	**33**	CLKOUT1
7	GPG0	**16**	GPG10	**25**	GND	**34**	GND
8	GPG6	**17**	nXDACK0	**26**	CLKOUT0	–	–
9	GPG1	**18**	GPG11	**27**	VDD33V	–	–

NOTES

2 TOOLKIT AND DEBUGGING

SMDK2410 ENVIRONMENT SETUP

The evaluation environments for the SMDK2410 are shown in Figure 2-1. The serial port (UART0) on the SMDK2410 has to be connected to COM port of the host PC. This can be used as a console for monitoring and debugging the SMDK2410. And the USB device on the SMDK2410 should be connected to the USB host of the host PC for downloading test images.

If you have an emulator such as MULTI-ICE and OPENice32-A900, you can use JTAG port on the SMDK2410 to interface the emulator.

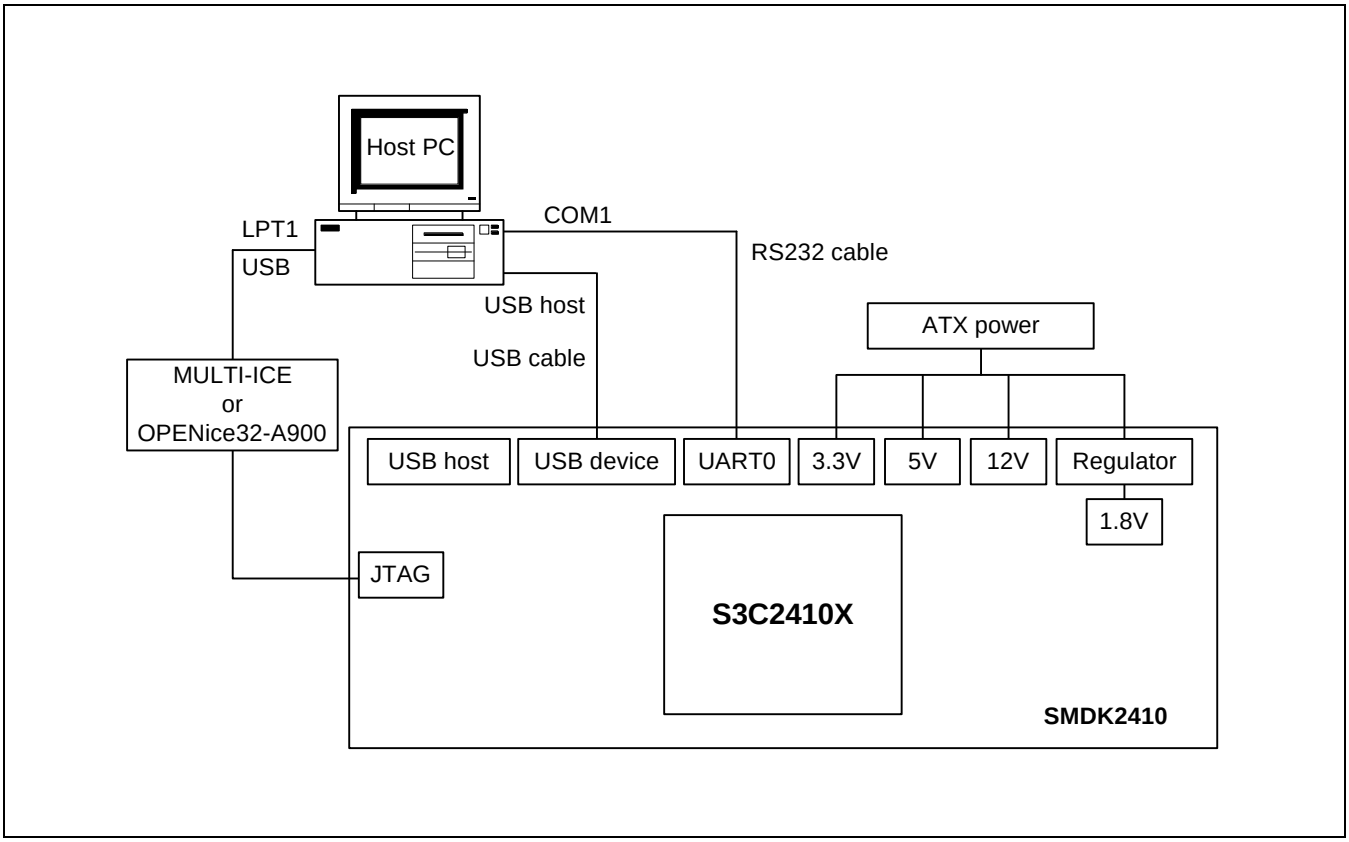


Figure 2-1. Setup Environment for SMDK2410 Board

RS232C CABLE CONNECTION

The serial cable is made as in Figure 2-2. The pins numbered only 1, 2, and 5 are used; make sure to check the cable's connections to prevent other pins from being used.

The UART0 and PC COM1 or COM2 port has to be connected through this cable connection.

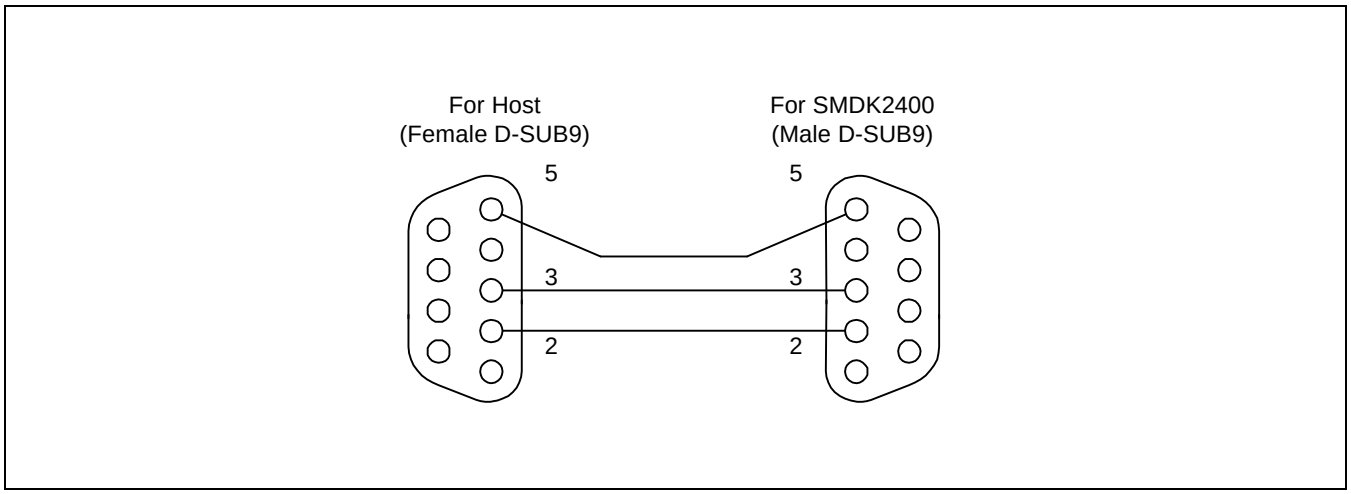


Figure 2-2. Serial Cable Connections for SMDK2410 Board

USB DOWNLOADER INSTALLATION (ON WINDOWS 98, ME, 2000 OR NT)

To install the USB downloader, follow the steps:

— Program the u241mon.bin into the flash memory of SMDK2410X board.

— Configure J35 & J36 to 1-2.

— Turn on the SMDK2410.

— If you installed the device driver for SMDK2400X/SMDK2410X before, overwrite new ‘secbulk.sys’ at C:\WINDOWS\SYSTEM32\DRIVERS. In this case, the step 6 will be skipped.

— Connect the SMDK2410X board with the PC (See Figure 2-3).

— When the USB device driver installation window appears, install the USB device driver (secbulk.inf). Note: ‘secbulk.inf’ and ‘secbulk.sys’ should be in the same directory (See Figure 2-4).

— Run ‘dnw.exe’.

— Turn the SMDK2410 off and then on.

— The message ([USB:OK]) in the window title bar indicates that the installation is successfully completed.

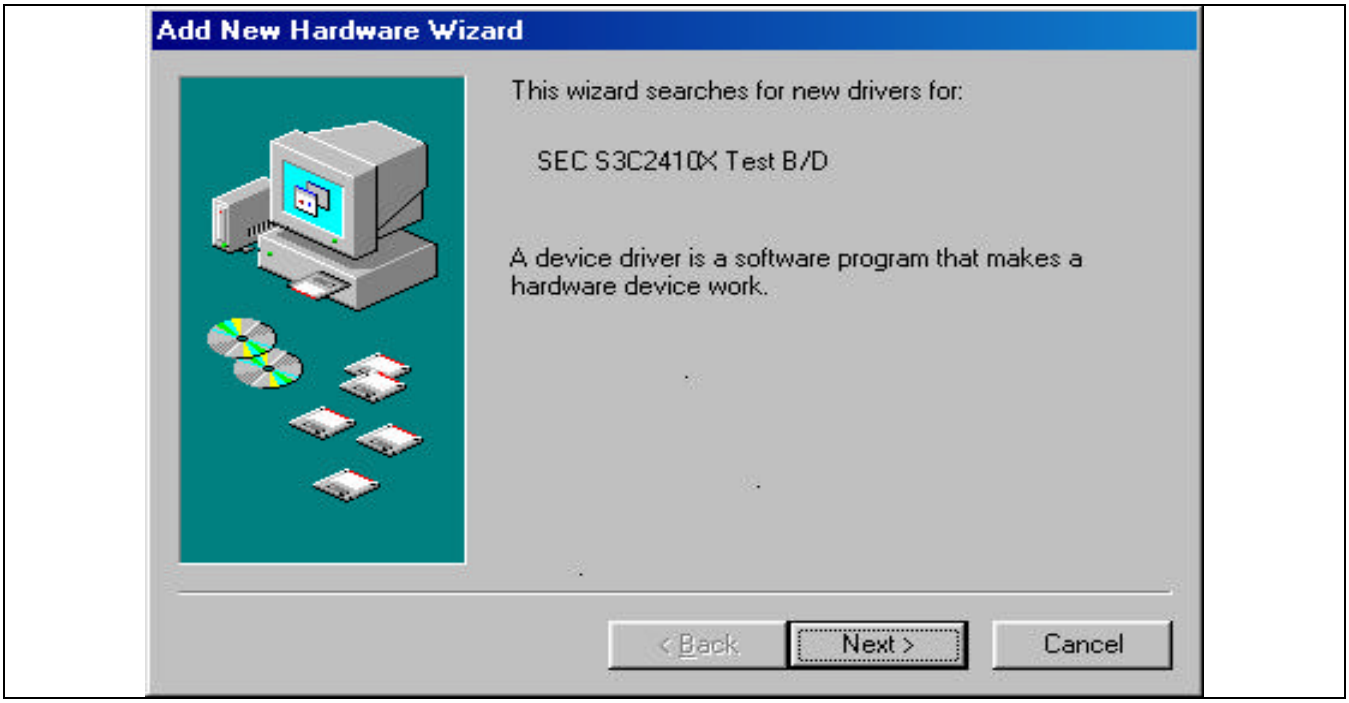


Figure 2-3. Add New Hardware Wizard (Window98)

NOTES:

1. If you have installed the device driver before, replace the old 'secbulk.sys' in C:\WINDOWS\SYSTEM32\DRIVERS with the new 'secbulk.sys'.
2. The maximum speed of the old 'secbulk.sys' with SMDK2400 is 490KB/S.
   New 'secbulk.sys' is optimized for SMDK2410 and the maximum speed will be about 770KB/S.
3. 'dnw.exe': PC USB/serial downloader program.
4. 'secbulk.inf' and 'secbulk.sys': PC USB driver.
5. 'u241mon.bin': S3C2410X USB downloader firmware.

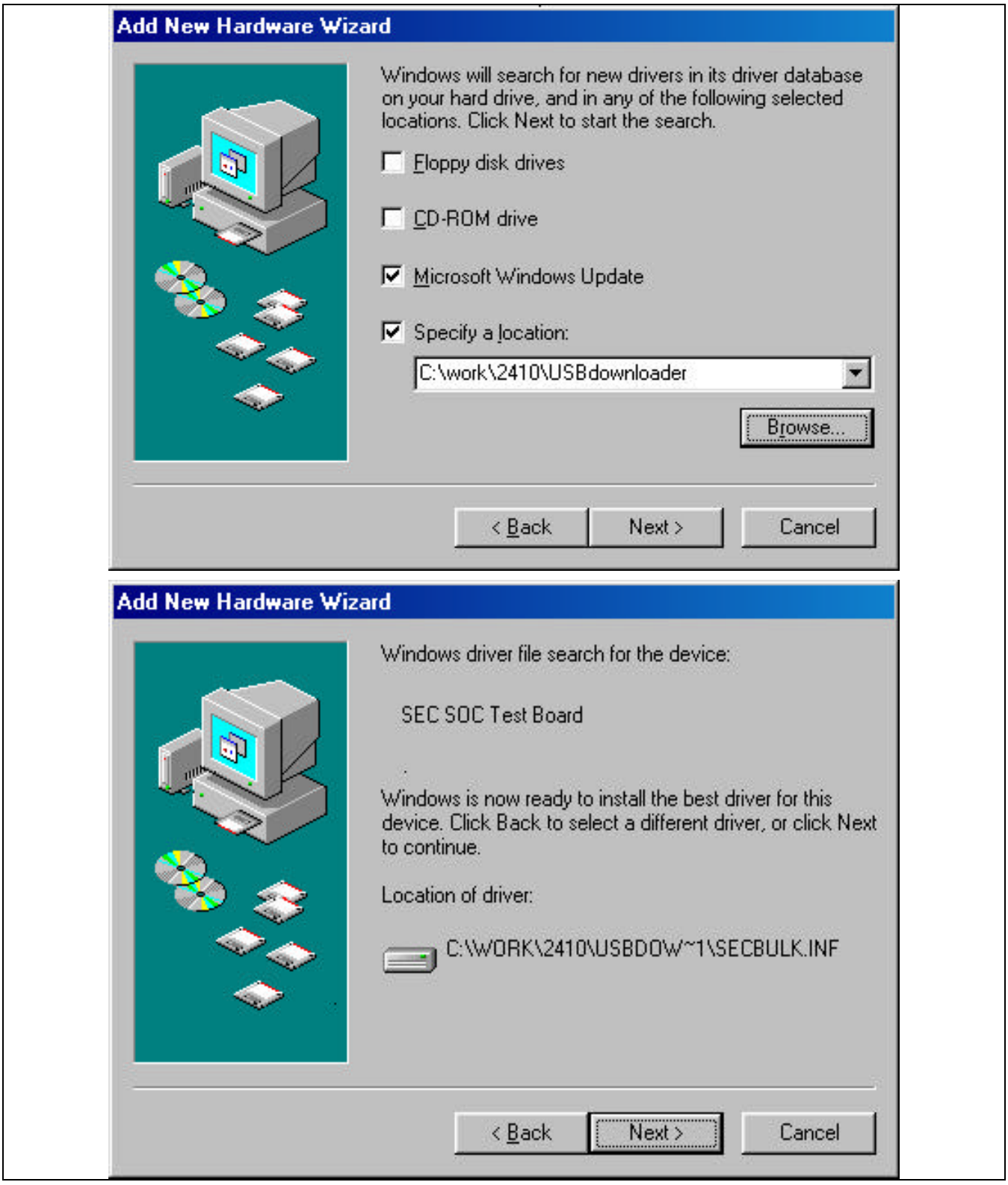


Figure 2-4. USB Device Driver Installation

Figure 2-4. USB Device Driver Installation (Continued)

CONFIGURING DNW

To configure the DNW, which works as USB and serial download utility, follow the steps:

1. Run the DNW.
2. Select Options from the Configuration menu (See Figure 2-5).
   Configuration → Options
3. Select Baud rate for serial communication.
   Serial communication properties of the DNW are as follows:
   Data bits:8-bit / Stop bits:1 / No flow control
4. Select a COM port of the host PC to communicate with the SMDK2410.
5. Set USB download address.
6. Click the OK button.

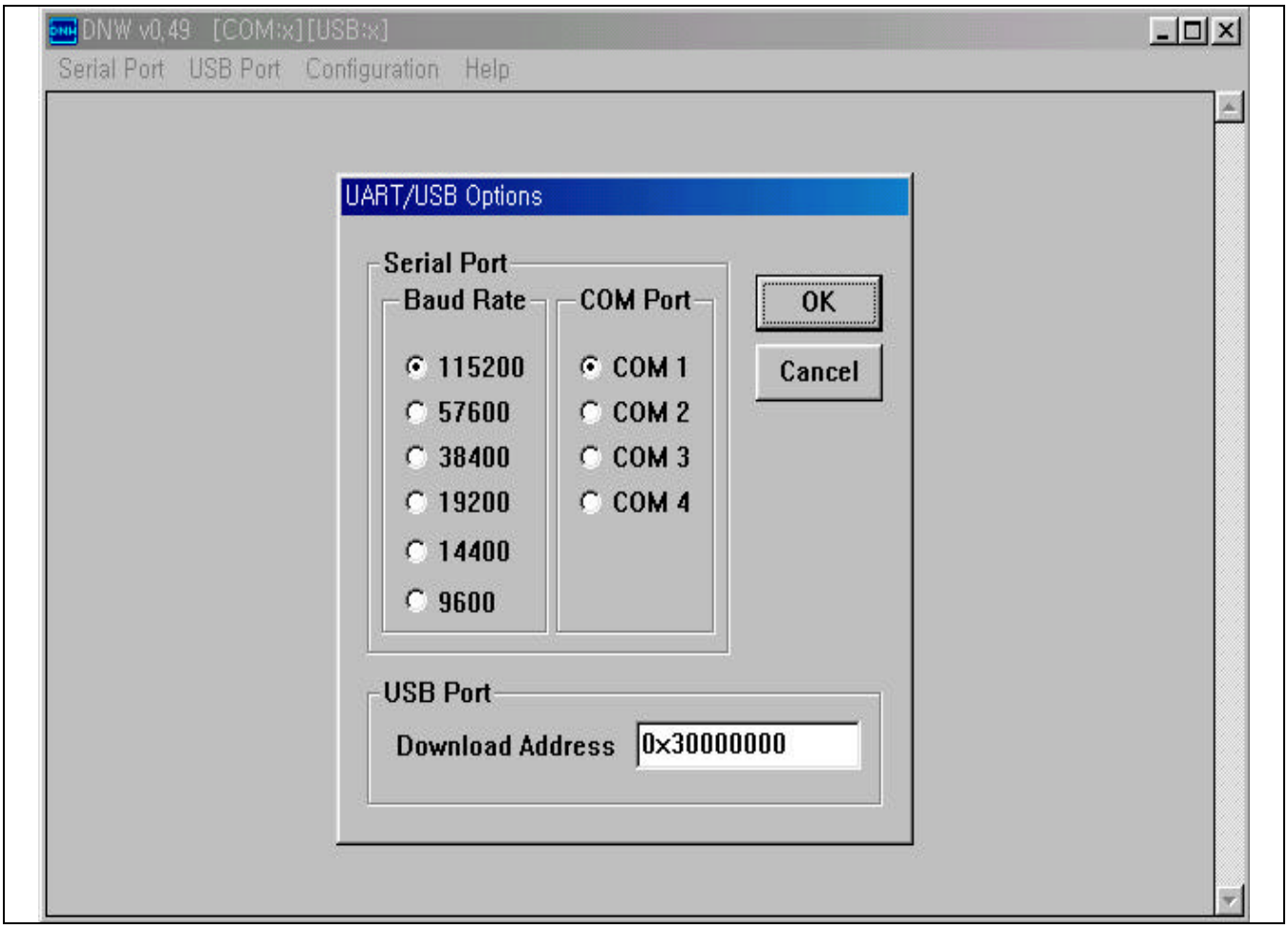


Figure 2-5. Setting UART/USB Options

CONNECT HOST PC AND SMDK2410 WITH DNW

After setting UART/USB options, users can activate UART and USB communication.

1. Select Connect from the Serial Port menu.
   Serial Port → Connect
2. Power on the SMDK2410 (See Figure 2-6).

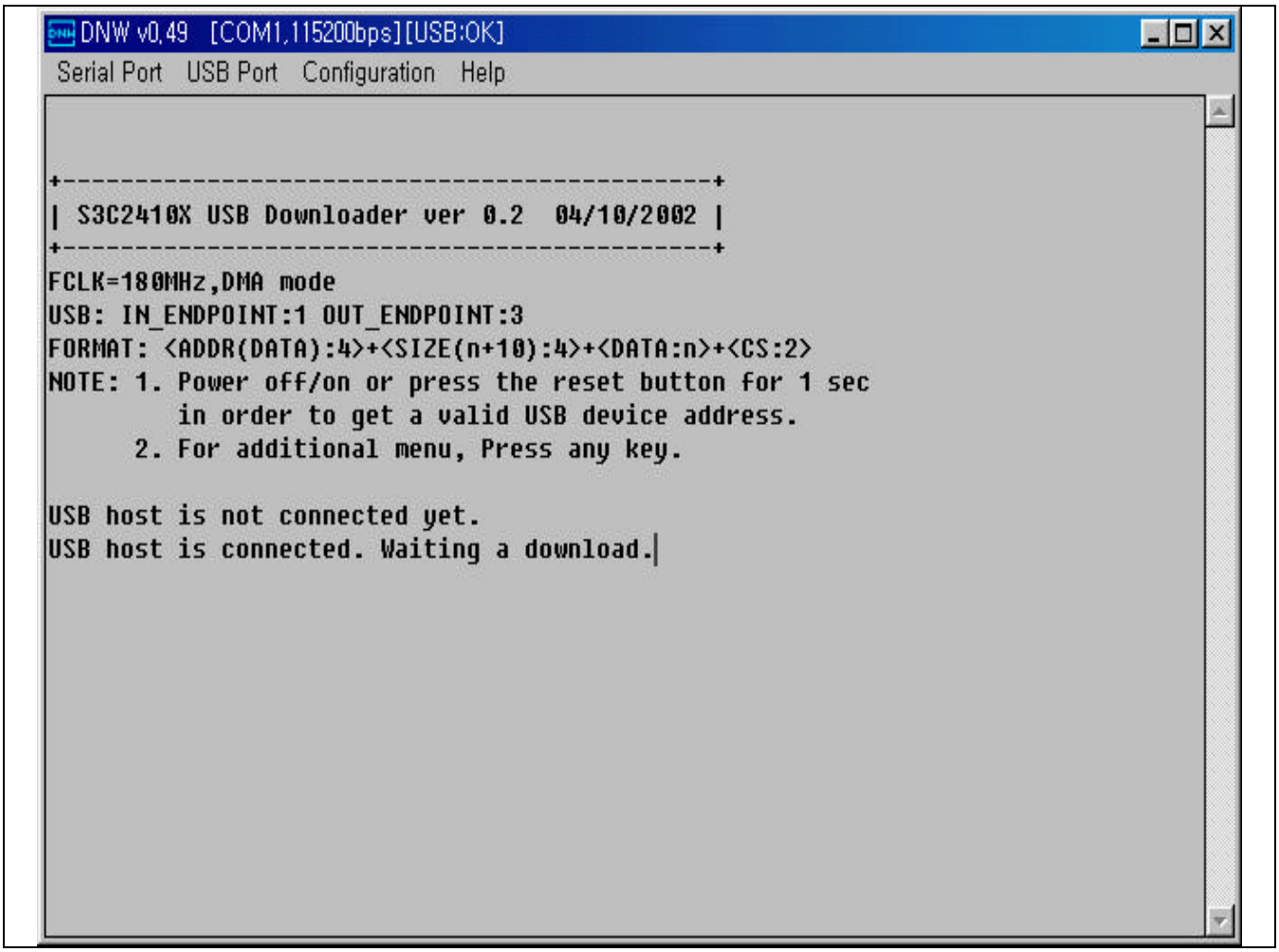


Figure 2-6. Power On Screen

INSTALL ARM TOOLKIT

First of all, install ARM toolkit 2.51, ADS (ARM Developer Suite) 1.0.1 or ADS1.1.

If you installed ARM toolkit 2.11a, then Makefile has to be changed a little. The toolkit 2.11a cannot support fromelf.exe utility. We recommend ARM Developer Suite 1.0.1, which is used in our development environment. We also recommend ADS 1.1.

The DOS environment variable has to be changed as follows after the installation of ARM toolkit 2.51.

SET ARMLIB=C:\ARM251\LIB\embedded
SET ARMINC=C:\ARM251\INCLUDE

HOW TO BUILD EXECUTABLE IMAGE FILE

Executable image file can be built by using the ARM Project manager or makefile. First, you have to build ELF format image (*.ELF or *.AXF). An ELF format image can be used for the ARM debugger directly. The binary file (.bin file) can be extracted from ELF format image.

First of all, you have to download S3C2410X evaluation source code and any other utilities from our web site (www.samsungsemi.com). They are helpful for you to understand the development environments of S3C2410X in an easier way. The distributed evaluation source code consists of following directories.

Directory	Description
BMP	Graphic header file converted from BMP file
obj	Object files
err	Error files

BUILDING 2410TEST.AXF (OR 2410TEST.ELF)

To build the sample source code, 2410TEST, run Makefile using the following commands.

cd 2410Test
armmake –a

or

cd 2410Test
make clean
make

After the procedure, 2410TEST.AXF (or 2410TEST.ELF) and 2410TEST.BIN image files will be seen in 2410TEST directory. The 2410TEST.AXF (or 2410TEST.ELF) file is used for ARM debugger.

The 2410TEST.BIN file is used for downloading through USB.

EXECUTING 2410TEST WITHOUT ARM MULTI-ICE OR OPENICE32-A900

First, U241MON has to operate on ROM. U241MON will be ready to receive 2410TEST.BIN. U241MON will launch 2410TEST.BIN after receiving 2410TEST.BIN.
To download 2410TEST.BIN through USB, after connecting the host PC and the SMDK2410 with the DNW follow the steps below:

1. Select Transmit from the USB Port menu.
   USB Port → Transmit
2. Select 2410TEST.BIN (See Figure 2-7).

Figure 2-7. 2410TEST Execution After its Downloading through USB

HOW TO USE ARM DEBUGGER WITH ARM MULTI-ICE

If you have built 2410TEST program without any error, you can find 2410TEST.AXF in 2410TEST directory. The generated image file will be downloaded to SDRAM memory on the SMDK2410 by ARM debugger through MDS like a MULTI-ICE. Next, you can start to debug the downloaded image using the ADW (ARM Debugger for Windows).

PREPARING AND CONFIGURING ARM MULTI-ICE

1. MULTI-ICE will be connected through JTAG port on the board. Connect all cables properly following its manual.
2. Start the ARM MULTI-ICE Server (Double click the MULTI-ICE Server icon).
3. Select Load Configuration from File menu and load 2400.CFG (See Figure 2-8).

   File → Load Configuration
4. Contents of 2400.CFG are as follows:

   [TITLE]
   S3C2400/S3C2410 TAP Configuration

   [TAP 0]
   ARM920T

   [Timing]
   Adaptive=OFF
5. Select Start-up Options from Settings menu.

   Settings → Start-up Options
6. Start-up Options dialog box is displayed (See Figure 2-9).

   Now you can select Load Configuration from Start-up Configuration and browse 2400.CFG
7. Start ARM Debugger using ARM debugger icon

   Also, you can start the debugger at DOS command window by typing adw 2410TEST.AXF.
   If you use ARM MULTI-ICE for the first time, you have to add Multi-ICE.DLL to ADW.
8. When ARM Debugger is started, it will load the image code to the Armulator (The Armulator is software emulator for ARM920T CPU).

If you ever used ARM Debugger as a remote debugger, Remote Debugger warning message dialog box will be displayed. If the remote debugger option is correct, select Yes; otherwise, select No.

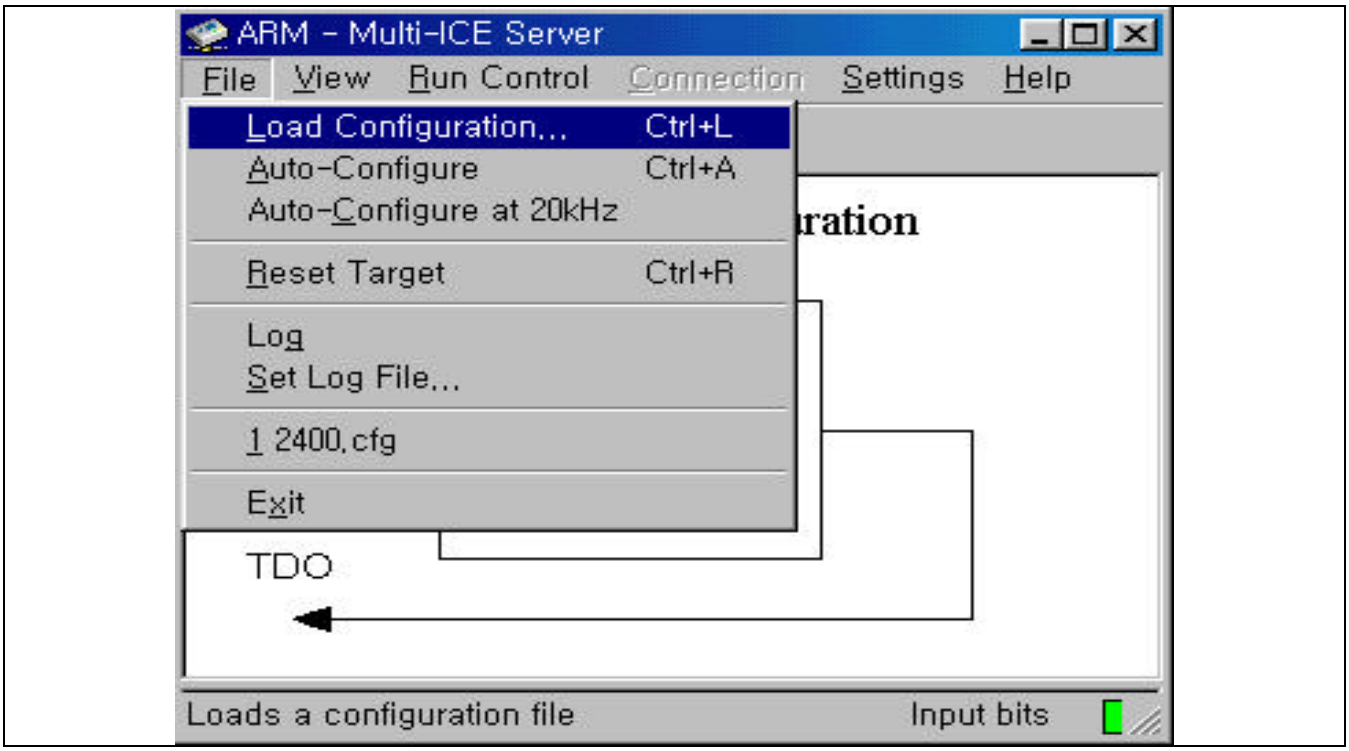


Figure 2-8. Load Configuration

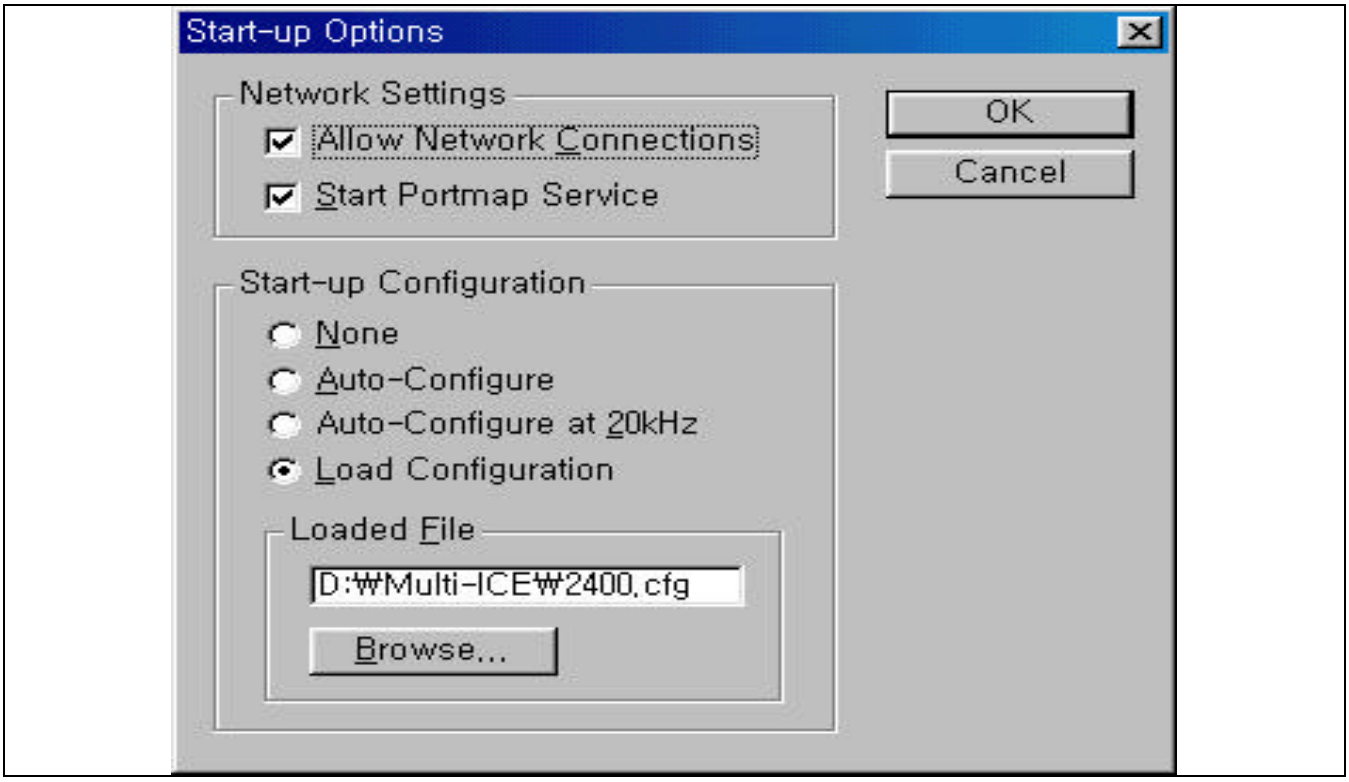


Figure 2-9. Start-up Configuration

CONFIGURING ARM DEBUGGER FOR ARM MULTI-ICE

In order to access a remote target, you should configure ARM Debugger for Windows (ADW). There are two kinds of ADW: one for Software Development Toolkit (SDT) and the other for ARM Developer Suit (ADS). These two kinds of ADW are basically same except some trivial differences. The below explanation will be described with ADW of ARM Developer Suit (ADS).
The MULTI-ICE interface unit must also be configured for the ARM core in the target system. The ARM920T core is contained in the S3C2410X on the SMDK2410 board.

To configure ARM Debugger using the MULTI-ICE interface, follow the steps:

1. Select Configure Debugger from the Options menu.

   Options -> Configure Debugger

   If you ever used ARM Debugger as a remote debugger, Remote Debugger warning message dialog box will be displayed. If the remote debugger option is correct, select Yes; otherwise, select No.

2. Debugger Configuration dialog box is displayed (See Figure 2-10).
   If there is no Multi-ICE in the target environment, then you have to select Add button and Multi-ICE.DLL.
   - ARMulator: lets you execute the ARM program without any physical ARM hardware by simulating ARM instructions in software.
   - Multi-ICE: connects the ARM debugger directly to the target board or to a MULTI-ICE unit attached to the target.

3. Select Multi-ICE from Target environment, and click the Configure button.

4. Configure ARM Multi-ICE dialog box (See Figure 2-11).
   - Connect page: select your host and MULTI-ICE communication target configuration.
   - Processor Settings page: set the cache clean code address.

5. Select Debugger page from Debugger Configuration dialog box (See Figure 2-12) and configure it.
   - Endian: little (If the big endian is used, Endian: big has to be selected.)

6. Select Memory Maps page from Debugger Configuration dialog box and configure it.
   - No Map File

7. If you click the OK button on Debugger Configuration dialog box, the debugger will be restarted. The restarting dialog box is displayed and numbers are rapidly changing, indicating that it is reading and writing to the target. This means that the executable image file is downloaded to the SDRAM code area.

This configuration is initially done and the setting is saved, which relieves the user of repeating another configuration next time.

Figure 2-10. Debugger Configuration: Target Page

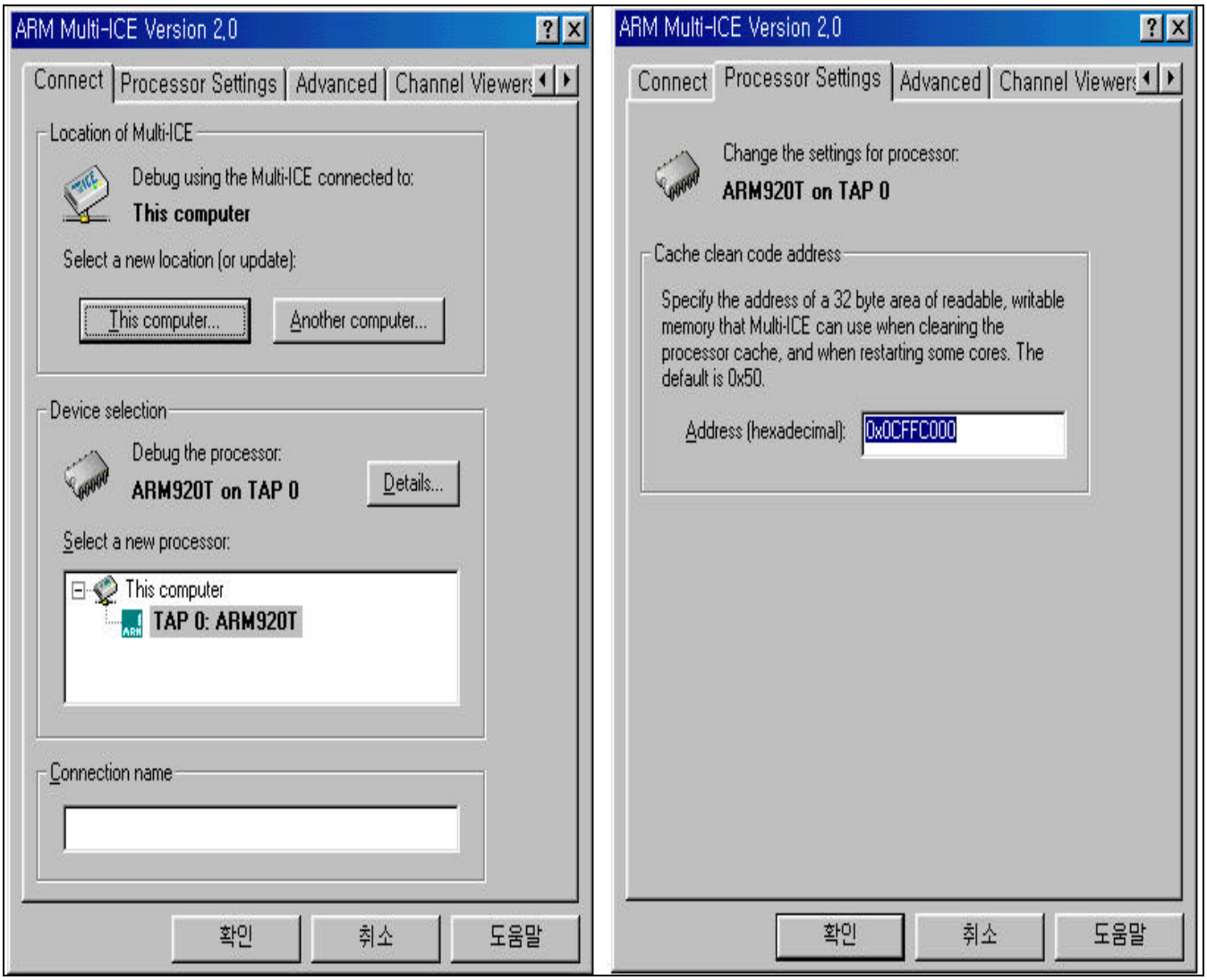


Figure 2-11. ARM Multi-ICE: Connect Page and Processor Settings Page

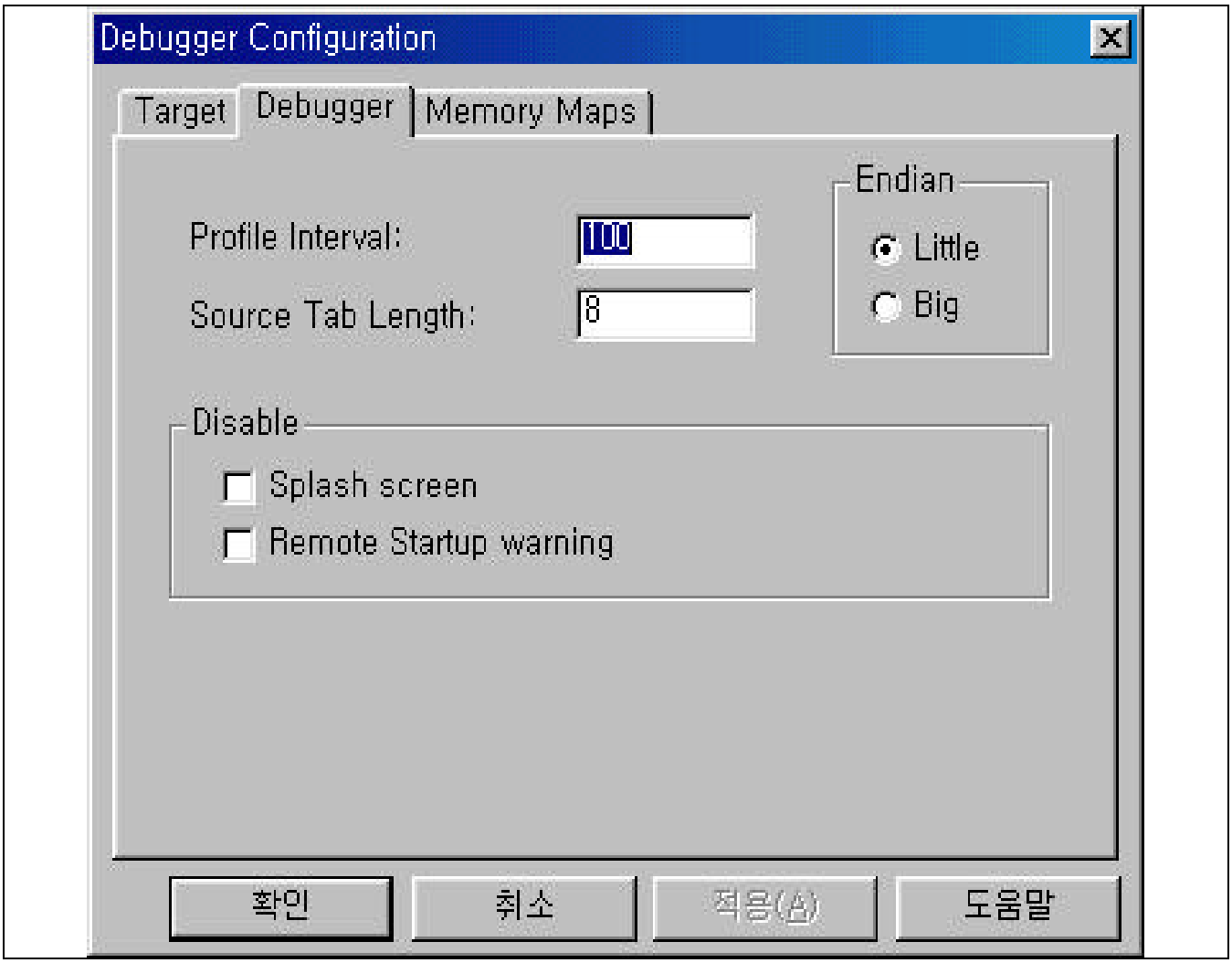


Figure 2-12. Debugger Configuration: Debugger Page

EXECUTING 2410TEST.AXF USING ARM MULTI-ICE

1. Initialize internal variables of the debugger. After a downloading, several windows are displayed, such as Execution window, Console window, and Command window. In Command window, you must initialize the internal variables of the debugger, "$semihosting_enabled" and "$vector_catch", by entering the following command:

   let $vector_catch = 0x00
   let $semihosting_enabled = 0x00 ;To use all H/W break points
   |SDT 2.51
   |let psr=%IFt_SVC32
   |ADW in ADS 1.0
   let psr=%IFt_SVC ;To disable all interrupts
   |break @0x0

Or, you can initialize these variables as follows:

First, create a text file named "armsd.ini", which includes the commands described above. Then, enter the following command in the Command window (See Figure 2-13):

obey c:\armutil\armsd.ini

For more information about these steps, refer to the reference document released by ARM.

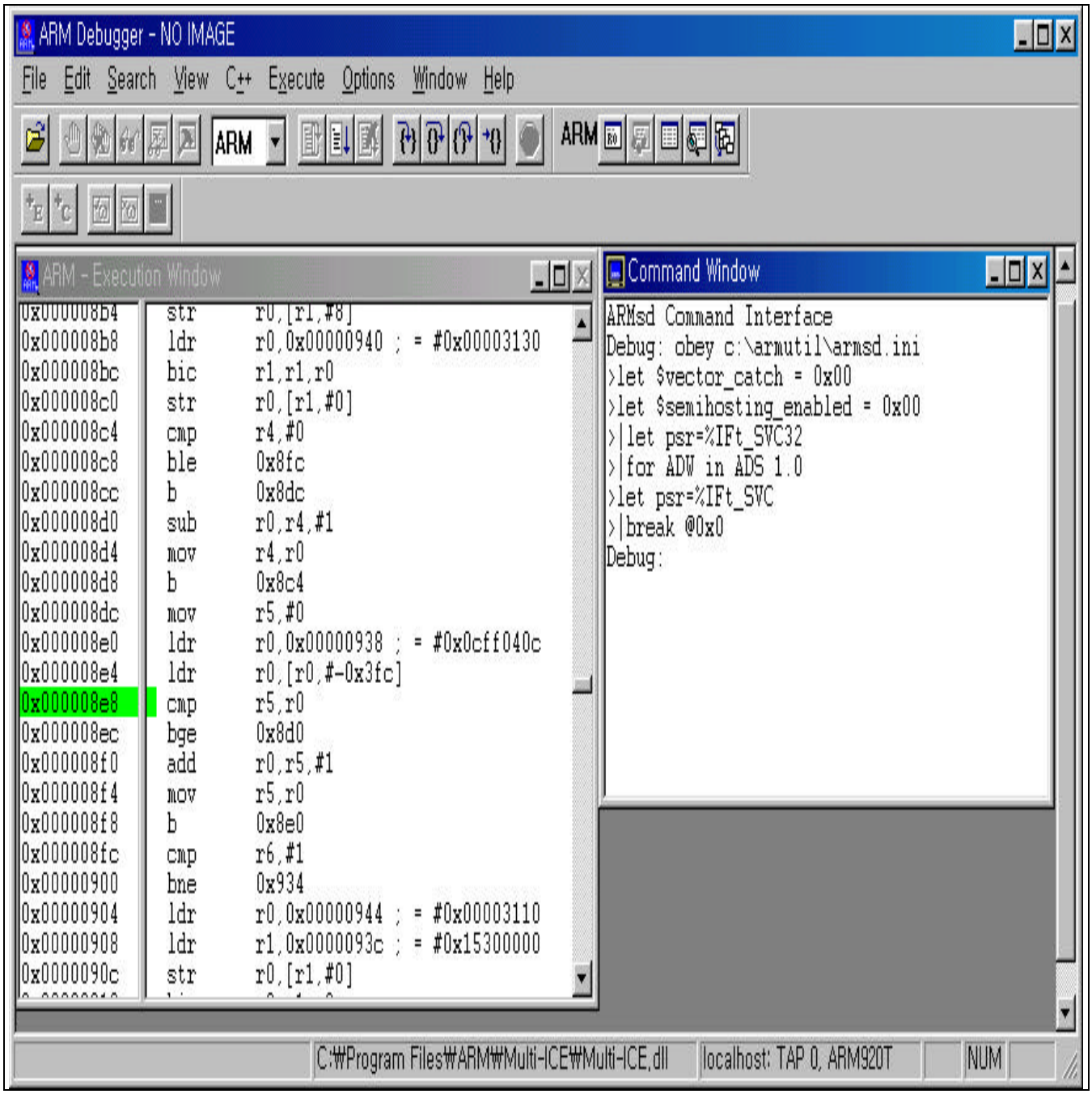


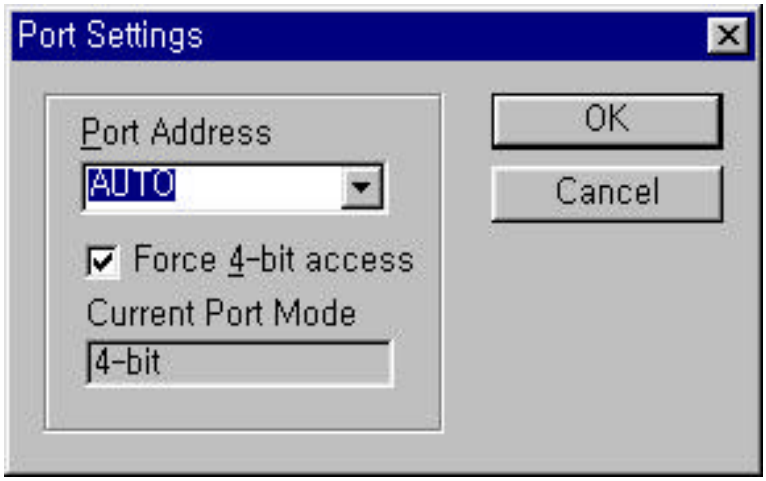
Figure 2-13. ARM Debugger Window (ADW): Command Window

2. Set breakpoint at Main in 2410TEST.c as follows:
   break Main
3. Execute the program by clicking Execute menu→Go. The program execution will stop at Main( ).
4. Now, the downloaded image file will run on SDRAM area. 2410TEST program running status can be monitored on the DNW.

MULTI-ICE CHECKPOINTS

1. Error messages

Refer to Error Messages of the Multi-ICE user's guide. If you cannot solve the problem by using the instructions in the user's guide, then apply the 'Force 4-bit access" option.



2. Multi-ICE current consumption problem

Multi-ICE draws the Multi-ICE operating current from a target board. The current is about 130mA at 3.3V. If the target board cannot supply the 130mA, an external power supply must be used for supplying the current to Multi-ICE.

3. nTRST, TMS, TCK and TDI pin connections

TMS, TCK and TDI pin must be pulled-up with 10K registers. If the Multi-ICE is not used when development is completed, nTRST must be 'L' level at least during the reset.

HOW TO USE ARM DEBUGGER WITH OPENICE32-A900

Followings explain how to download the compiled image to SDRAM memory on the SMDK2410 by ARM debugger through OPENice32-A900, an emulator for ARM processor.

Note:

If you need technical support of OPENice32-A900, please contact AIJI System (http://www.aijisystem.com, openice@aijisystem.com)

CONFIGURING ARM DEBUGGER FOR OPENice32-A900

To debug the target board with OPENice32-A900, you should configure ARM Debugger for Windows (ADW) or ARM Extended Debugger (AxD). As MULTI-ICE, OPENice32-A900 should be connected through JTAG port on the board and switched on.

To configure ARM Debugger for OPENice32-A900 interface, follow the steps:

1. Select Configure Debugger from the Options menu.

   Options -> Configure Debugger

   If you ever used ARM Debugger as a remote debugger, Remote Debugger warning message dialog box will be displayed. If the remote debugger option is correct, select Yes; otherwise, select No.

2. Debugger Configuration dialog box is displayed (See Figure 2-14). If there is no OPENice32-A900 in the target environment box, then you have to click on the Add button and select OPENice32-A900.DLL.
   - ARMulator: lets you execute the ARM program without any physical emulator by simulating ARM instructions in software.
   - OPENice32-A900: connects the ARM debugger to OPENice32-A900 attached to the target board.

3. Select OPENice32-A900 from the Target environment box, and click the Configure button.

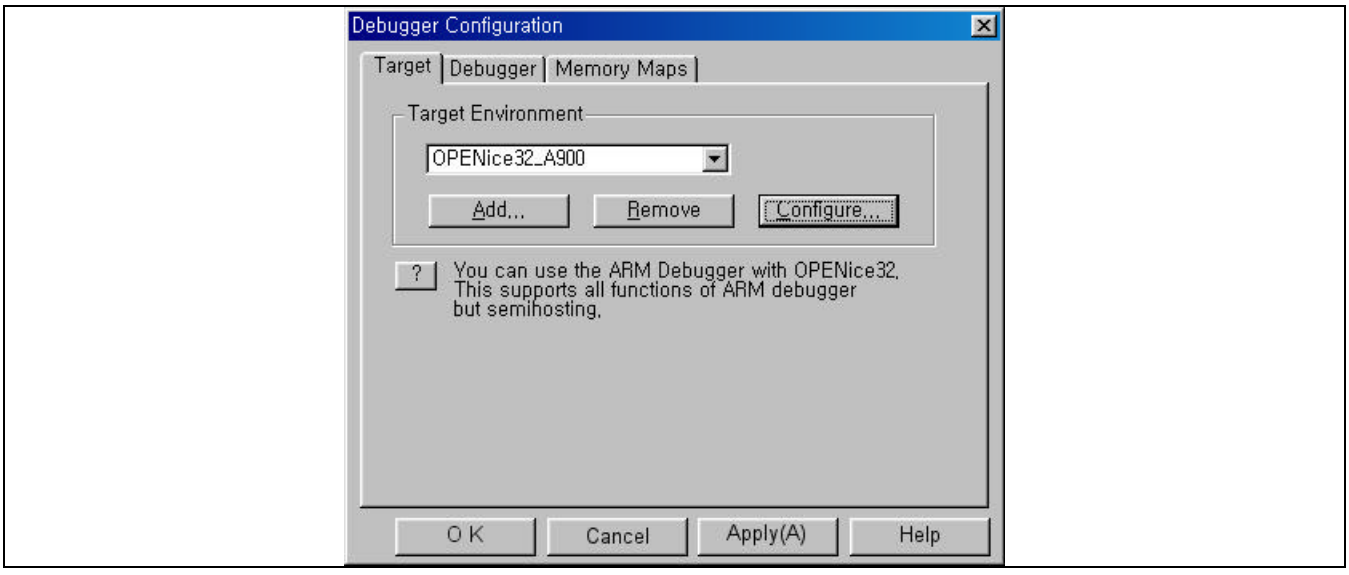


Figure 2-14. Debugger Configuration: Target Page

4. ConfigureOPENice32-A900 dialog box (See Figure 2-15, 2-16).
   — Remote page: select the connection to OPENICE32-A900.

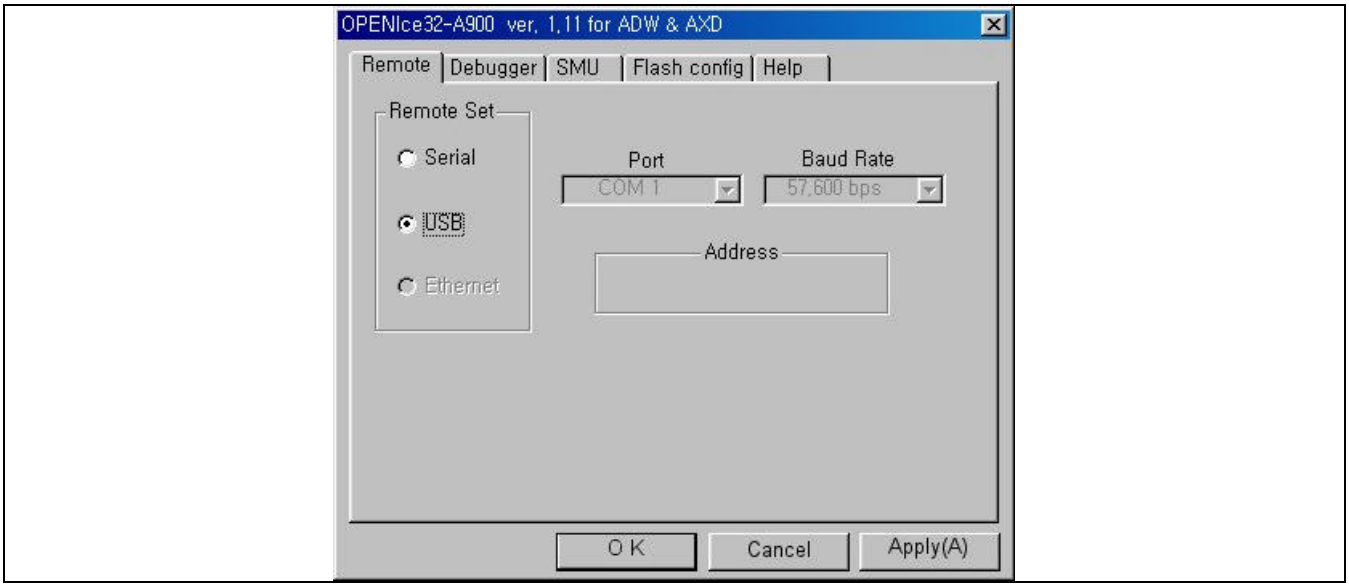


Figure 2-15. OPENice32-A900 Configuration: Communication Setting Page

— Debugger page : set the Endian and decide where initializes S3C2410X without any boot ROM.
Endian : little (If the big endian is used, Endian: big has to be selected.).
It should be matched with the option that you set in the compiler.
To init SMU : If you want to initialize S3C2410X on the board without any boot ROM, check it and set SMU in the SMU page, (Don't check it in this application.)
Flash download : If image file will be downloaded to a flash device, check it and set options in the Flash config page. (Don't check it in this application.)

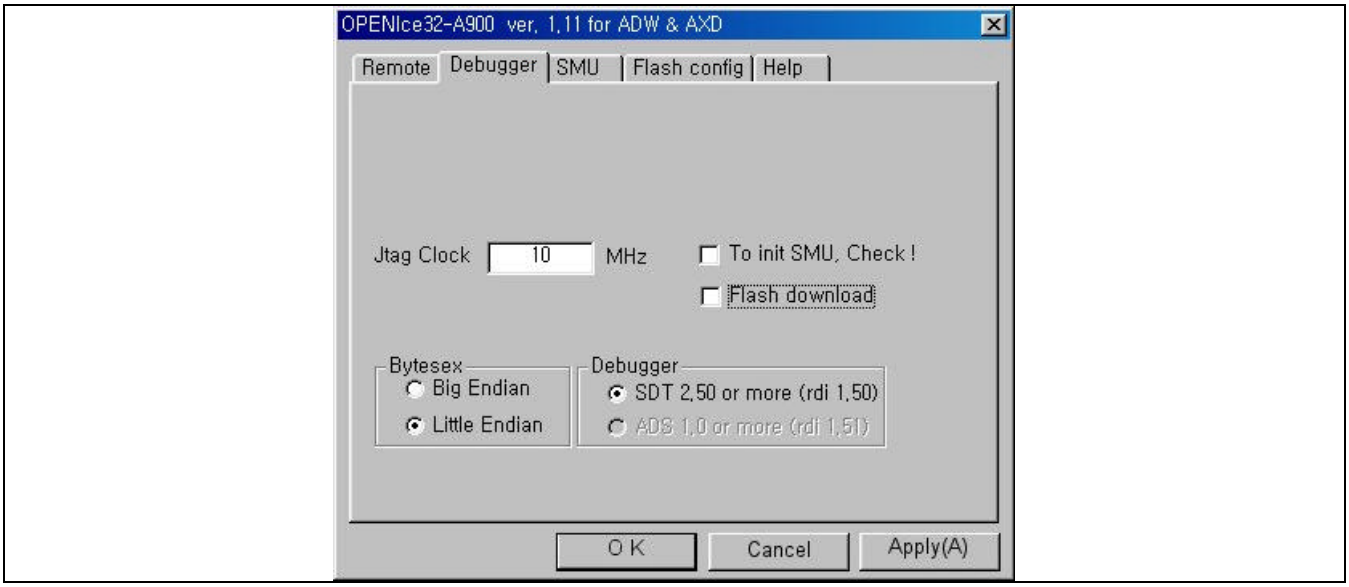


Figure 2-16. OPENice32-A900 Configuration: Endian and SMU Setting Page

— SMU page: set SMU of S3C2410X. Select SMDK2410 or S3C2410X from the device name and modify the values.(No need to set it in this application).

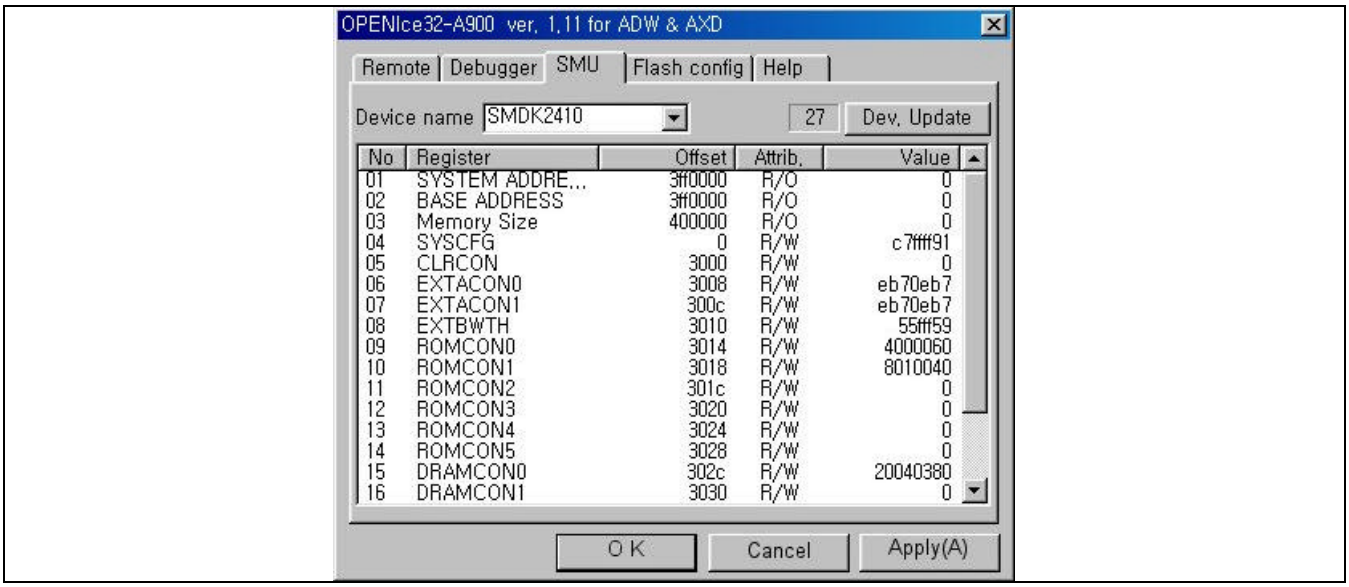


Figure 2-17. OPENice32-A900 Configuration: Communication Setting Page

— Flash Config page: set options for Flash download. (No need to set it in this application).

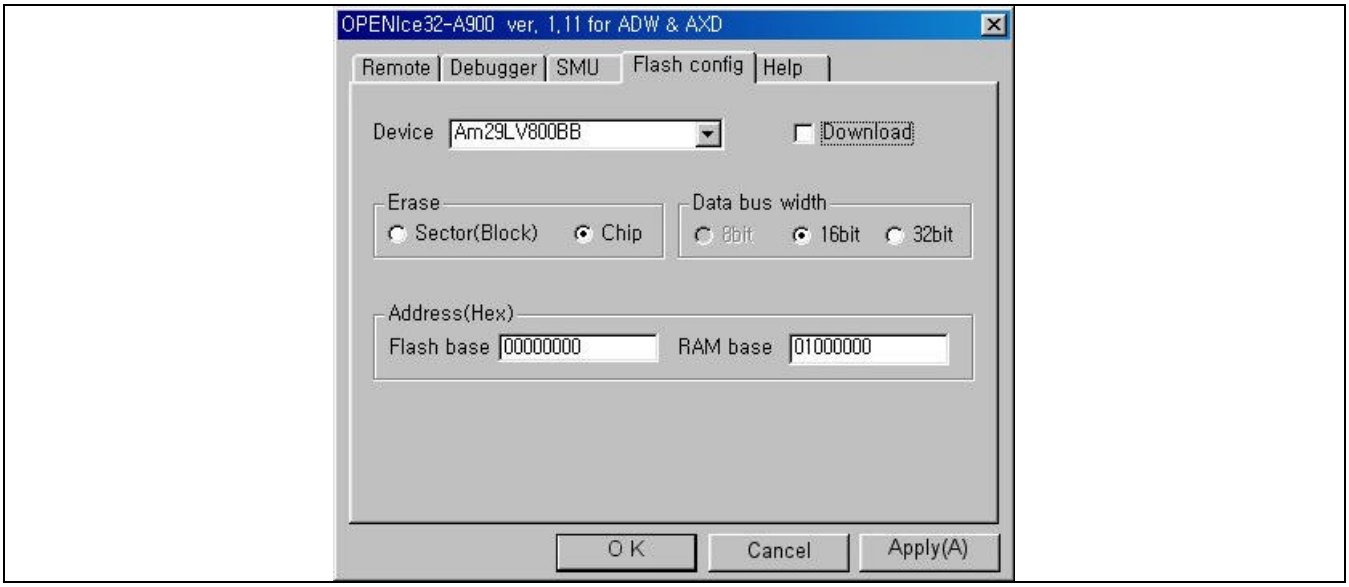


Figure 2-18. OPENice32-A900 Configuration: Endian and SMU Setting Page

5. Select Debugger page from Debugger Configuration dialog box (See Figure 2-19) and configure it
   — Endian: little (If the big endian is used, Endian: big should be selected.). It should be matched with the option that you set in the compiler.

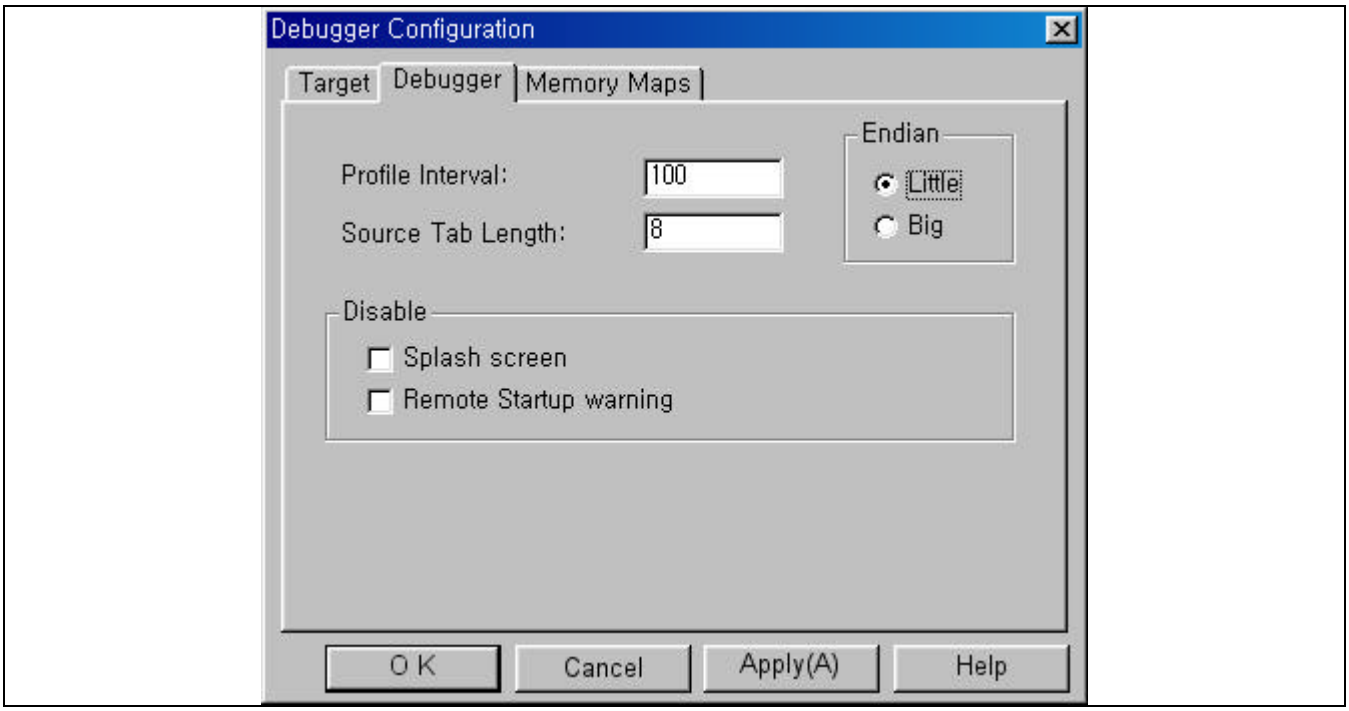


Figure 2-19. Debugger Configuration: Debugger Page

6. Select Memory Maps page from Debugger Configuration dialog box and configure it.
   — No Map File

7. If you click the OK button on Debugger Configuration dialog box, the debugger will be restarted. The restarting dialog box is displayed and numbers are rapidly changing, indicating that it is reading and writing to the target. This means that the executable image file is downloaded to the SDRAM code area.

This configuration is initially done and the setting is saved, which relieves the user of repeating another configuration next time.

EXECUTING 2410TEST.AXF USING OPENICE32-A900

1. Select Load Image from the File menu and select the compiled image (2410TEST.AXF). Then it will be downloaded to the SDRAM on the board.
2. Execute the program by select Go from the Execute menu.
3. Now, the downloaded image file will run on SDRAM area. 2410TEST program running status can be monitored on the DNW.

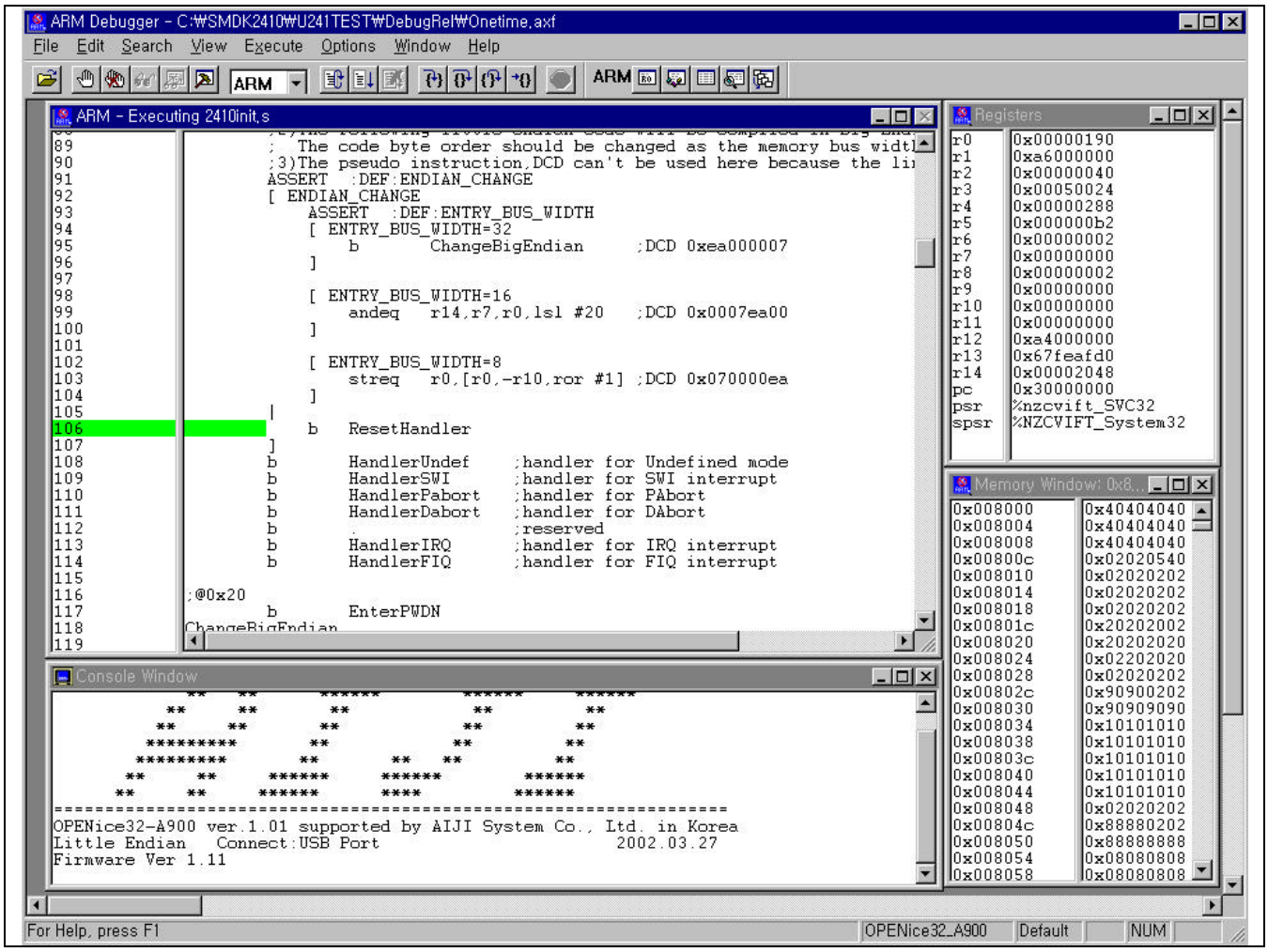


Figure 2-20. ARM Debugger Window (ADW): After Downloading

DEBUGGING DOWNLOADED IMAGE IN ADW

Stepping Through Program

To step through the program execution flow, you can select one of the following three options:

— Step: advances the program to the next line of code that is displayed in the execution window.

— Step Into: advances the program to the next line of code that follows all function calls. If the code is in a called function, the function source is displayed in the Execution window and the current code.

— Step Out: advances the program from the current function to the point from which it was called immediately after the function call. The appropriate line of code is displayed in the Execution window.

Setting Breakpoint

A breakpoint is the point you set in the program code where the ARM debugger will halt the program operation. When you set a breakpoint, it appears as a red marker on the left side of the window.

To set a simple breakpoint on a line of code, follow these steps:

1. Double-click the line where you want to place a break, or choose Toggle Breakpoint from the Execute menu. The Set or Edit Breakpoint dialog box is displayed.
2. Set the count to the required value or expression (The program stops only when this expression is correct).

To set a breakpoint on a line of code within a particular program function:

1. Display a list of function names by selecting Function Names from View menu.
2. Double-click the function name you want to open. A new source window is displayed containing the function source.
3. Double-click the line where the breakpoint is to be placed, or choose Toggle Breakpoint from the Execute menu. The Set or Edit Breakpoint dialog box appears.
4. Set the count to the required value or expression (The program stops only when this expression is correct).

Setting Watch Point

A watch point halts a program when a specified register or a variable, which is set to a specific number, is about to be changed.

To set a watch point, follow these steps:

1. Display a list of registers, variables, and memory locations you want to watch by selecting the Registers, Variables, and Memory options from the View menu.
2. Click the register, variable, or memory area in which you want to set the watch point. Then, choose Set or Edit Watchpoint from the Execute menu.
3. Enter a Target Value in the Set or Edit Watchpoint dialog box. Program operation will stop when the variable reaches the specified target value.

VIEWING VARIABLES, REGISTERS, AND MEMORY

You can view and edit the value of variables, registers, and memory by choosing the related heading from the View menu:

— Variables: for global and local variables.

— Registers: for the current mode and for each of the six register view modes.

— Memory: for the memory area defined by the address you enter.

DISPLAYING CODE INTERLEAVED WITH DISASSEMBLY

If you want to display the source code interleaved with disassembly, choose Toggle Interleaving on the Options menu. This command toggles between Displaying Source Only and Displaying Source Interleaved with Disassembly. When the source code is shown interleaved with disassembly, machine instructions appear in a lighter gray color.

For additional information about ARM Debugger, refer to the reference document released by ARM.

SWITCHING DEVELOPMENT TOOLKIT

USB boot code (U241mon.c) and test code (2410test.c) can be executed in the SDT or ADS by changing the option in OPTION.H and Makefile. In other words, boot and test code can be translated from ADS to SDT and vice versa by changing option in the following table.

Table 2-1. Toolkit Switching Options

	ADS	SDT
Makefile	fromelf -nodebug -bin -output $(PRJ).bin $(PRJ).elf	fromelf -nodebug -nozeropad $(PRJ).elf -bin $(PRJ).bin
OPTION.H	#define ADS10 TRUE	#define ADS10 FALSE

TRANSLATING CODE FROM ADS INTO SDT

U241MON and 2410TEST codes were optimized for ADS 1.0. In other words, these codes were compiled and linked by the ADS 1.0. So, these codes should be modified to work on the SDT.
If you want to compile our codes with the SDT, then you have to change the definition of ADS10 in OPTION.H from 'TRUE' to 'FALSE' and the option in makefile from 'fromelf -nodebug -bin -output $(PRJ).bin $(PRJ).elf' to 'fromelf -nodebug -nozeropad $(PRJ).elf -bin $(PRJ).bin' option.

TRANSLATING CODE FROM SDT INTO ADS

First function __rt_lib_init(); is applied to the main code. And then old Makefile for SDT is changed to a new one for ADS.
If you have used SDT 2.50, it is recommended that you should read related documents (ADS, Getting Started, and ARM DUI0064A) about the difference between SDT 2.50 and ADS 1.0.

REMOVED OR CHANGED ITEMS FROM MAKEFILE FOR SDT 2.50

1. ARMLINK option
   - first: the path of an object file is not needed.
2. ARMASM option
   - cpu: should be changed as -cpu ARM920T
   - apcs: should be changed to -apcs /noswst
3. Compiler option
   - fc : should be removed.
   - zpz0 : should be removed. This is not needed any more.
   - apcs : should be changed to -apcs /noswst
   - processor : should be removed.
   - arch : should be removed.
   - cpu : should be added as -cpu ARM920T
4. fromelf.exe
   - nozeropad: should be removed. This is not needed any more.
   - output : command line style should be changed using -output option as follows:
     fromelf -nodebug -bin -output $(BIN)\$(PRJ).bin $(BIN)\$(PRJ).AXF

OTHER ITEMS SHOULD BE CHANGED FOR ADS 1.0

1. ammake.exe
   The armmake.exe is not supplied with ADS 1.0. So, you have to use your own Make utility. (nmake.exe, make.exe, pmake.exe, or armmake.exe in SDT 2.50).
2. Embedded library
   There is no separate embedded library in ADS 1.0. All the library in ADS 1.0 is made for embedded applications.
   But, the library must be initialized using __rt_lib_init() function. If you do not use __rt_lib_init(), the C library does not work well.
3. There is no tasm.exe. The tasm.exe is merged into armasm.exe.

EXAMPLE OF MAKEFILE FOR ADS 1.0

This is a sample makefile on ADS 1.0.

```
##### File Definition ####
PRJ = 2410test
INIT= 2410init
AM1 = 2410slib
AM2 = 2410swis
CM1 = 2410lib
CM2 = mmu
CM3 = 2410iis
CM4 = timer
CM5 = 2410RTC
CM6 = 2410IIC
        .
        .
        .
CM38 = spi
CM39 = strata32

#### Destination path Definition ####
OBJ=.\obj
ERR=.\err

#### ARM tool Definition ####
ARMLINK = armlink
ARMASM = armasm
ARMCC   = armcc

#### Option Definition ####
LFLAGS = -ro-base 0x30000000 -elf -map -xref \
         -list list.txt -first $(INIT).o(Init)
AFLAGS = -li -apcs /noswst -cpu ARM920T
CFLAGS = -c -g+ -li -apcs /noswst -cpu ARM920T
#### Object combine Definition ####

OBJS =$(OBJ)\$(INIT).o $(OBJ)\$(AM1).o $(OBJ)\$(AM2).o $(OBJ)\$(PRJ).o \
        $(OBJ)\$(CM1).o $(OBJ)\$(CM2).o $(OBJ)\$(CM3).o $(OBJ)\$(CM4).o \
        $(OBJ)\$(CM5).o $(OBJ)\$(CM6).o $(OBJ)\$(CM7).o $(OBJ)\$(CM8).o \
        $(OBJ)\$(CM9).o $(OBJ)\$(CM10).o $(OBJ)\$(CM11).o $(OBJ)\$(CM12).o \
        $(OBJ)\$(CM13).o $(OBJ)\$(CM14).o $(OBJ)\$(CM15).o $(OBJ)\$(CM16).o \
        $(OBJ)\$(CM17).o $(OBJ)\$(CM18).o $(OBJ)\$(CM19).o $(OBJ)\$(CM20).o \
        $(OBJ)\$(CM21).o $(OBJ)\$(CM22).o $(OBJ)\$(CM23).o $(OBJ)\$(CM24).o \
        $(OBJ)\$(CM25).o $(OBJ)\$(CM26).o $(OBJ)\$(CM27).o $(OBJ)\$(CM28).o \
        $(OBJ)\$(CM29).o $(OBJ)\$(CM30).o $(OBJ)\$(CM31).o $(OBJ)\$(CM32).o \
        $(OBJ)\$(CM33).o $(OBJ)\$(CM34).o $(OBJ)\$(CM35).o $(OBJ)\$(CM36).o \
        $(OBJ)\$(CM37).o $(OBJ)\$(CM38).o $(OBJ)\$(CM39).o
```

```
all: $(PRJ).axf

clean:
        del $(OBJ)\*.o

$(PRJ).axf: $(OBJS)
        del $(PRJ).bin
        del $(PRJ).axf
        $(ARMLINK) $(LFLAGS) -o $(PRJ).axf $(OBJS)
        fromelf -nodebug -bin -output $(PRJ).bin $(PRJ).axf

#For SDT2.5 fromelf -nodebug -nozeropad $(PRJ).elf -bin $(PRJ).bin
#For ADS1.0 fromelf -nodebug -bin -output $(PRJ).bin $(PRJ).elf


$(OBJ)\$(PRJ).o:  $(PRJ).c  2410addr.h  2410lib.h  makefile
        del $(OBJ)\$(PRJ).o
        del $(ERR)\$(PRJ).err
        $(ARMCC) $(CFLAGS)  $(PRJ).c -o $(OBJ)\$(PRJ).o -Errors $(ERR)\$(PRJ).err
        .
        .
        .
        .
        .
$(OBJ)\$(CM27).o:  $(CM27).c  2410addr.h  2410lib.h  makefile
        del $(OBJ)\$(CM27).o
        del $(ERR)\$(CM27).err
        $(ARMCC) $(CFLAGS)  $(CM27).c -o $(OBJ)\$(CM27).o -Errors $(ERR)\$(CM27).err

$(OBJ)\$(CM28).o:  $(CM28).c  2410addr.h  2410lib.h  makefile
        del $(OBJ)\$(CM28).o
        del $(ERR)\$(CM28).err
        $(ARMCC) $(CFLAGS)  $(CM28).c -o $(OBJ)\$(CM28).o -Errors $(ERR)\$(CM28).err

$(OBJ)\$(CM29).o:  $(CM29).c  2410addr.h  2410lib.h  makefile
        del $(OBJ)\$(CM29).o
        del $(ERR)\$(CM29).err
        $(ARMCC) $(CFLAGS)  $(CM29).c -o $(OBJ)\$(CM29).o -Errors $(ERR)\$(CM29).err
        .
        .
        .
        .
        .
```

3 PROGRAMMING FLASH MEMORIES

PROGRAMMING NAND FLASH MEMORY

The SMDK2410 supports NAND flash control interface. There are two methods to write images to NAND flash memory:

— Write image files to NAND flash memory with write-program.

— Write image files to NAND flash memory with JTAG interface.

NAND FLASH WRITE WITH WRITE-PROGRAM

The target image must be downloaded in SDRAM before executing write-program.

To download and write a target image from the host to SDRAM through USB interface, follow the steps:

1. Run the DNW utility program (See Figure 3-1).

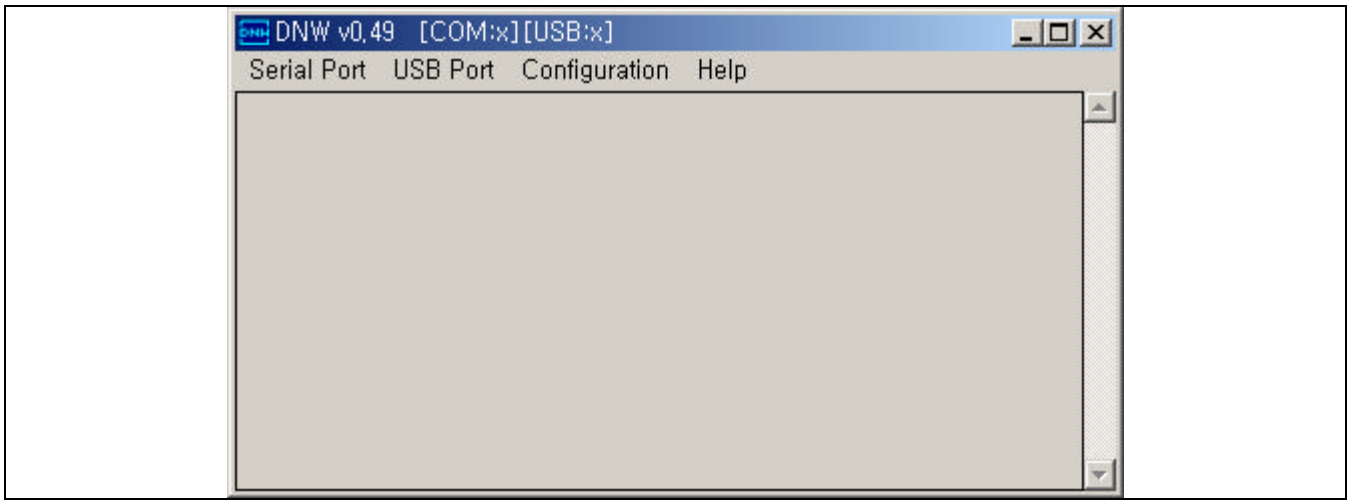


Figure 3-1. DNW Window to Download

2. Select Serial Port on the system menu of DNW and click Connect to open the serial port (See Figure 3-2, 3).

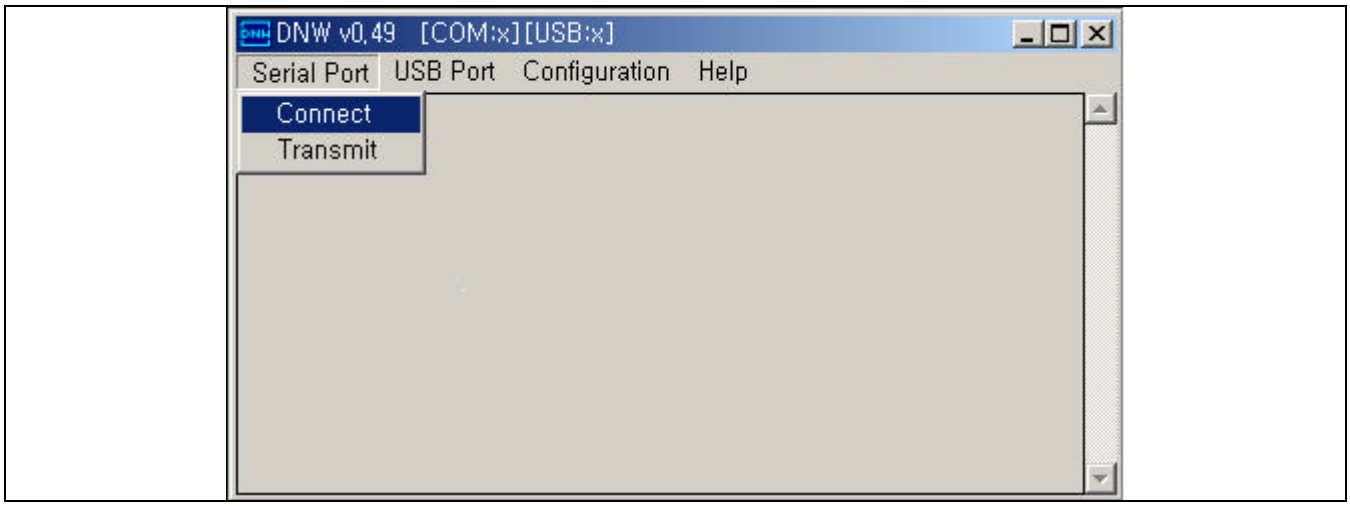


Figure 3-2. DNW Window (to Connect Serial Port)

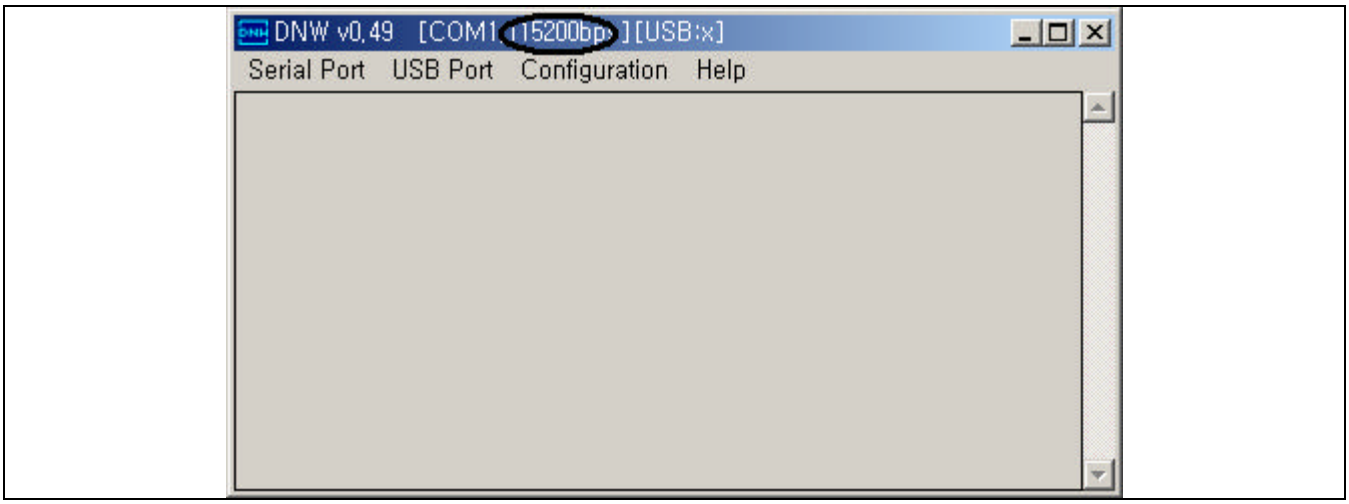


Figure 3-3. DNW Window (after Open Baud-rate is Printed on Title Bar)

3. Connect the serial and USB cable from the host PC to SMDK2410 system and turn on the power of SMDK2410 board (See Figure 3-4).

NOTES:

1. Connector J33 must be open.
2. SMDK2410 must run monitor program that is provided by SAMSUNG.

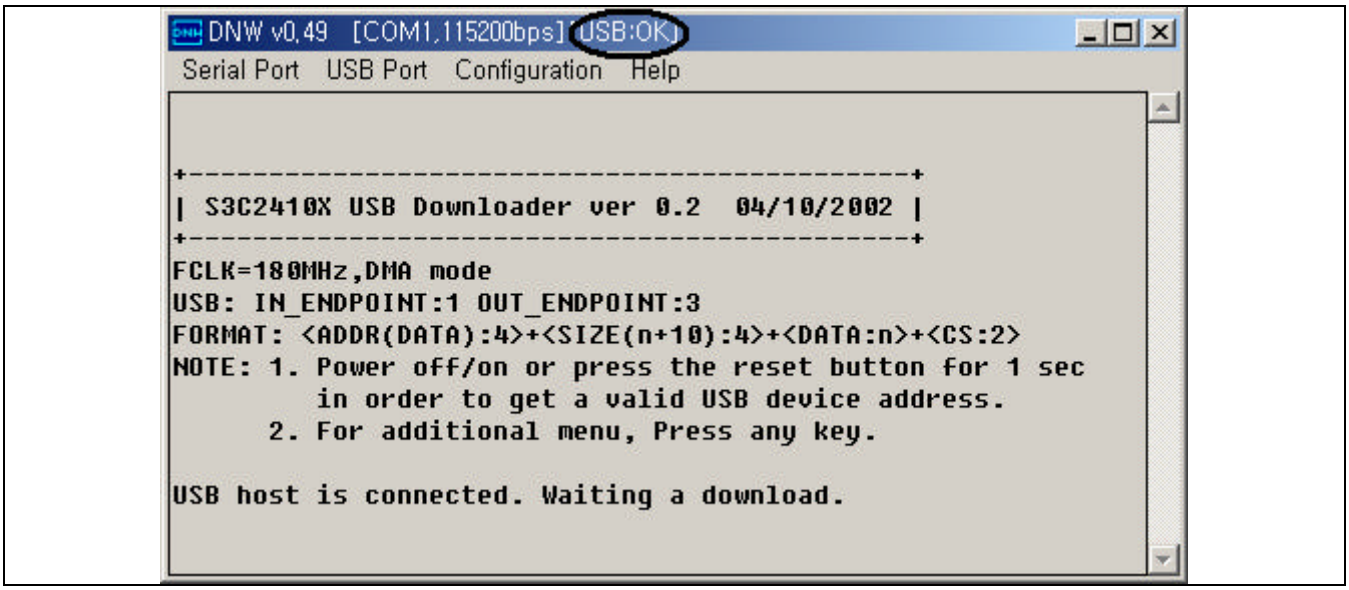


Figure 3-4. DNW Window (after Turning on the SDMK2410)

4. To see the additional menu, press any key on the DNW window (See Figure 3-5).

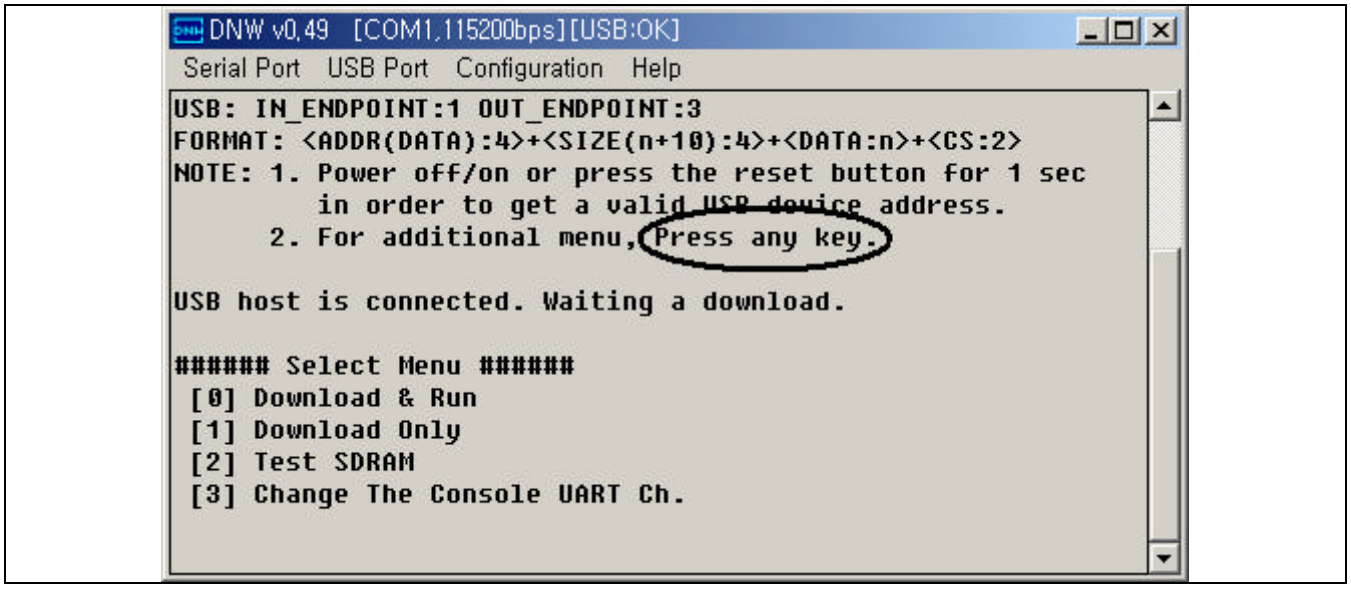


Figure 3-5. DNW Window to Download

5. For downloading a target image, select Download Only item on the DNW window (See Figure 3-6).
6. Write the address to download and press enter key (See Figure 3-6).

NOTE: The target image must be located on 0x30100000 address.

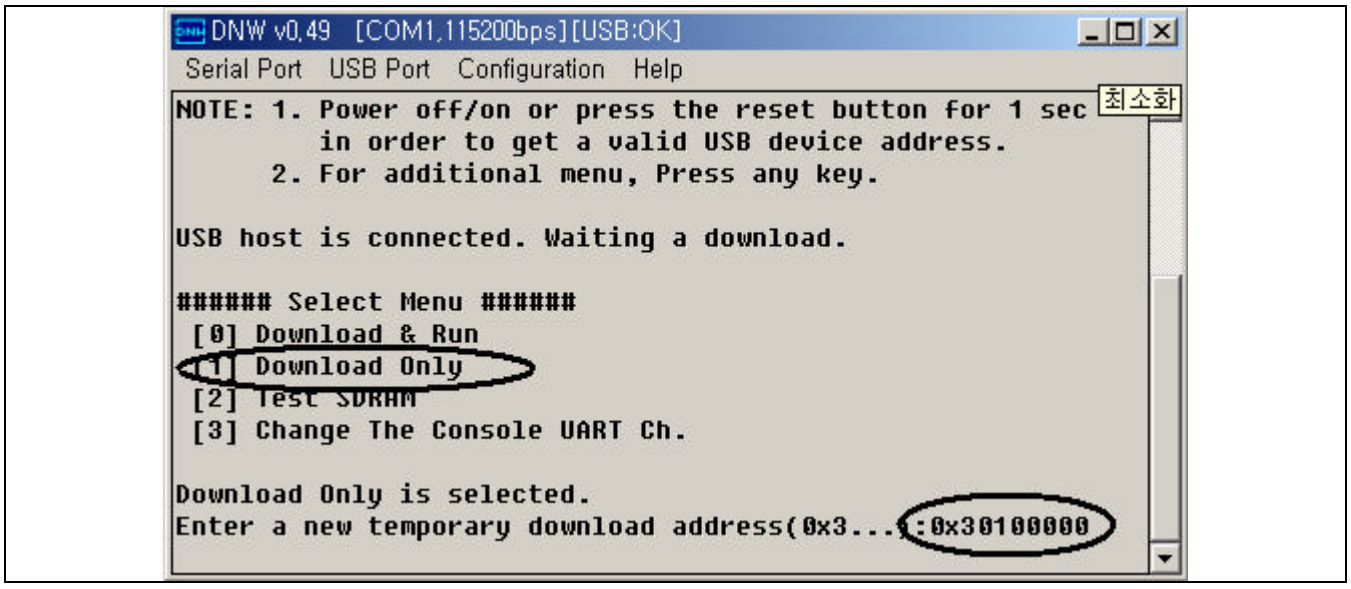


Figure 3-6. DNW Window (to Select Target Image)

7. Select USB Port on the system menu of the DNW and click Transmit to download a target image (See Figure 3-7).

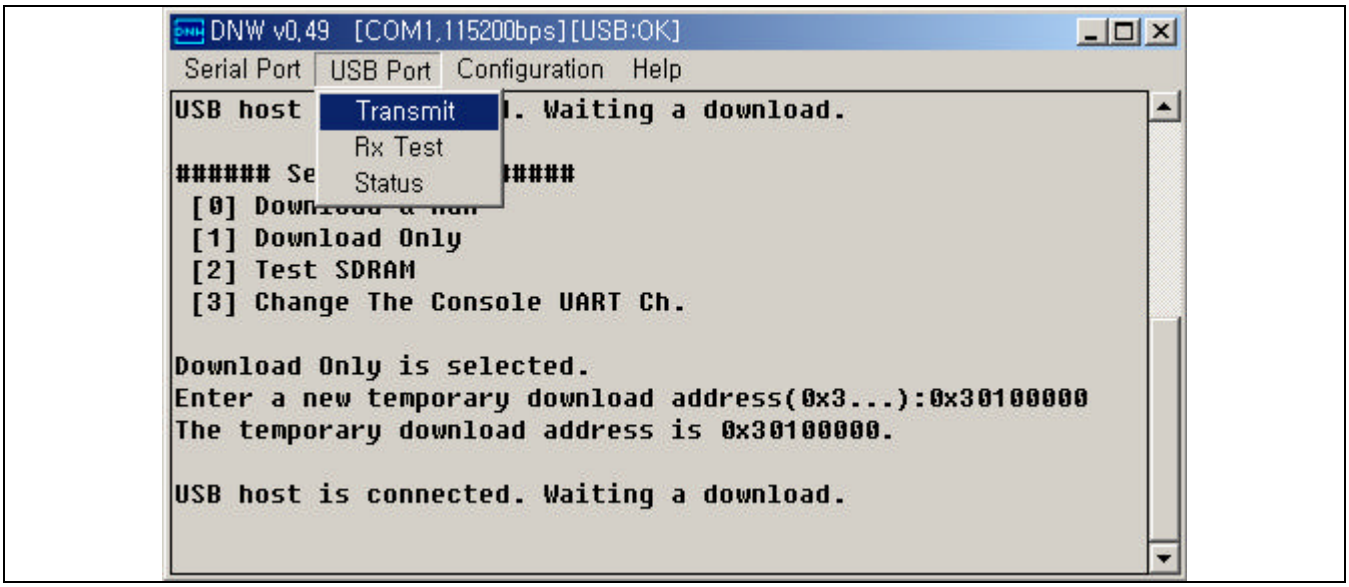


Figure 3-7. DNW Window (for USB Downloading)

8. Select a target image on file open dialog box (See Figure 3-8).

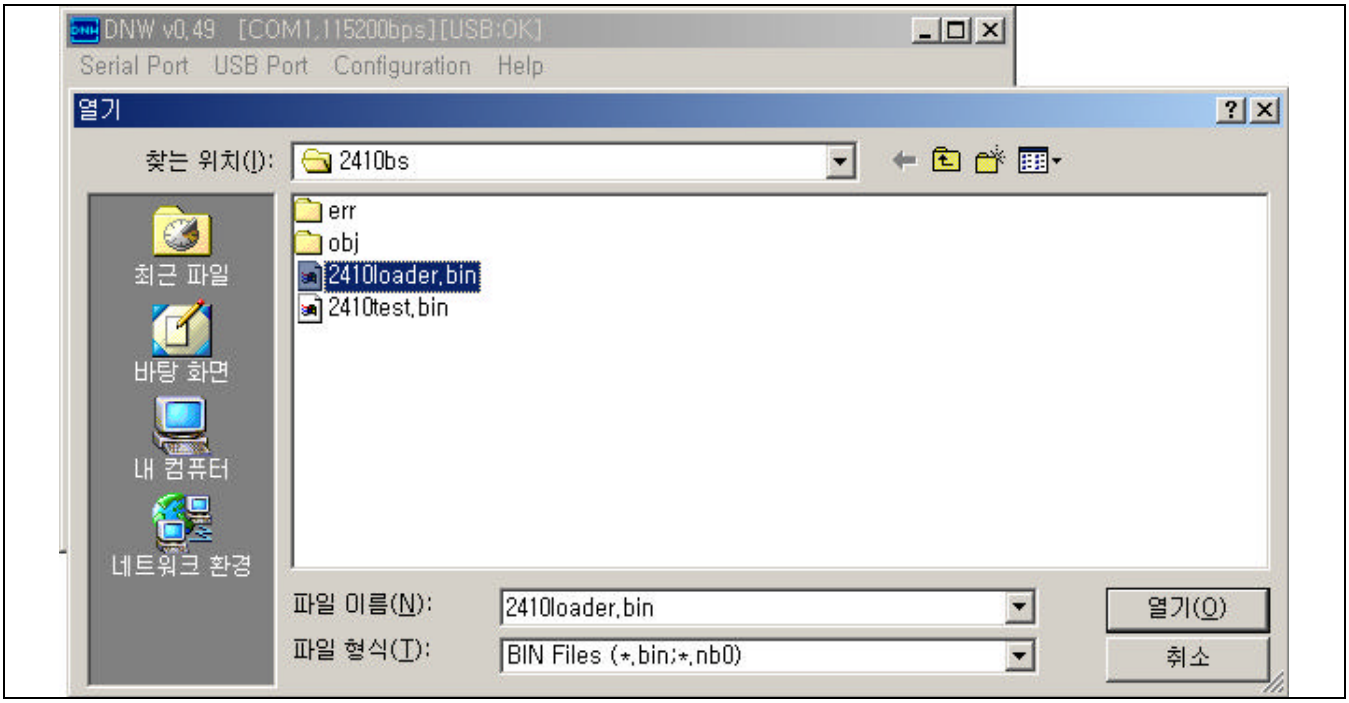


Figure 3-8. DNW Window (File Open Dialog Box)

9. For downloading 2410test program, select Download & Run item on the DNW window and download 2410test program like step 7 (See Figure 3-9).

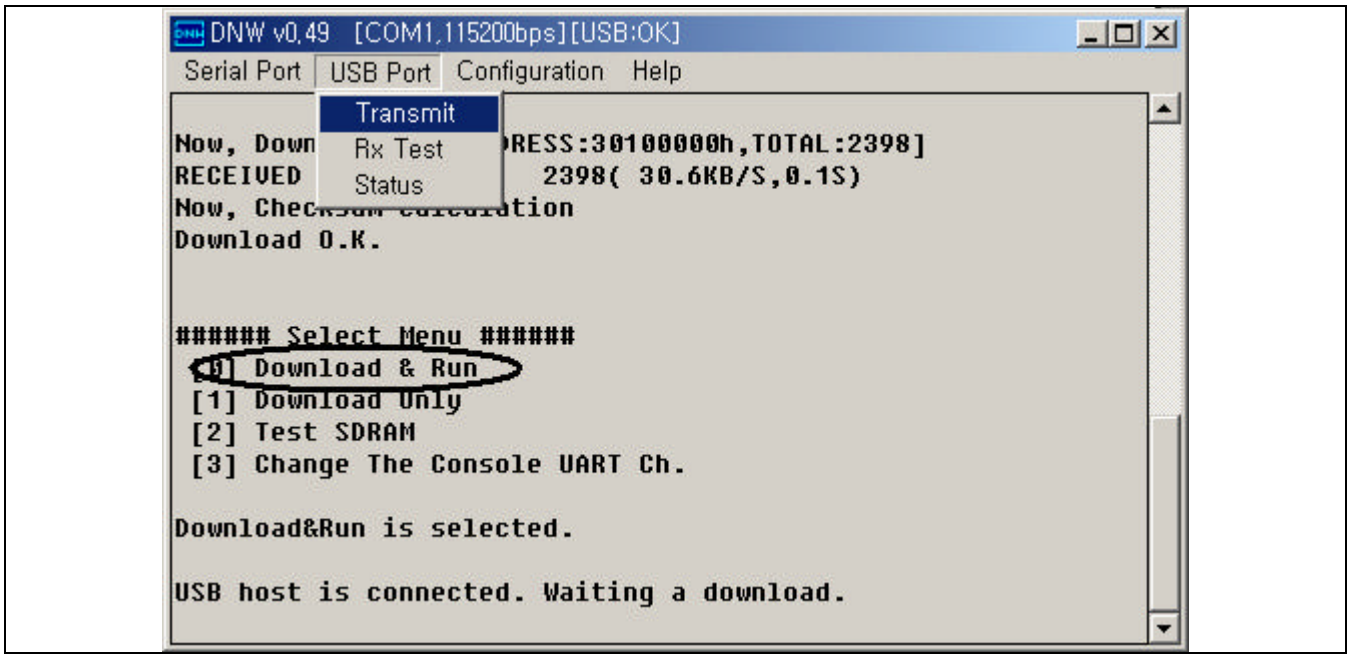


Figure 3-9. DNW Window (File Open Dialog Box)

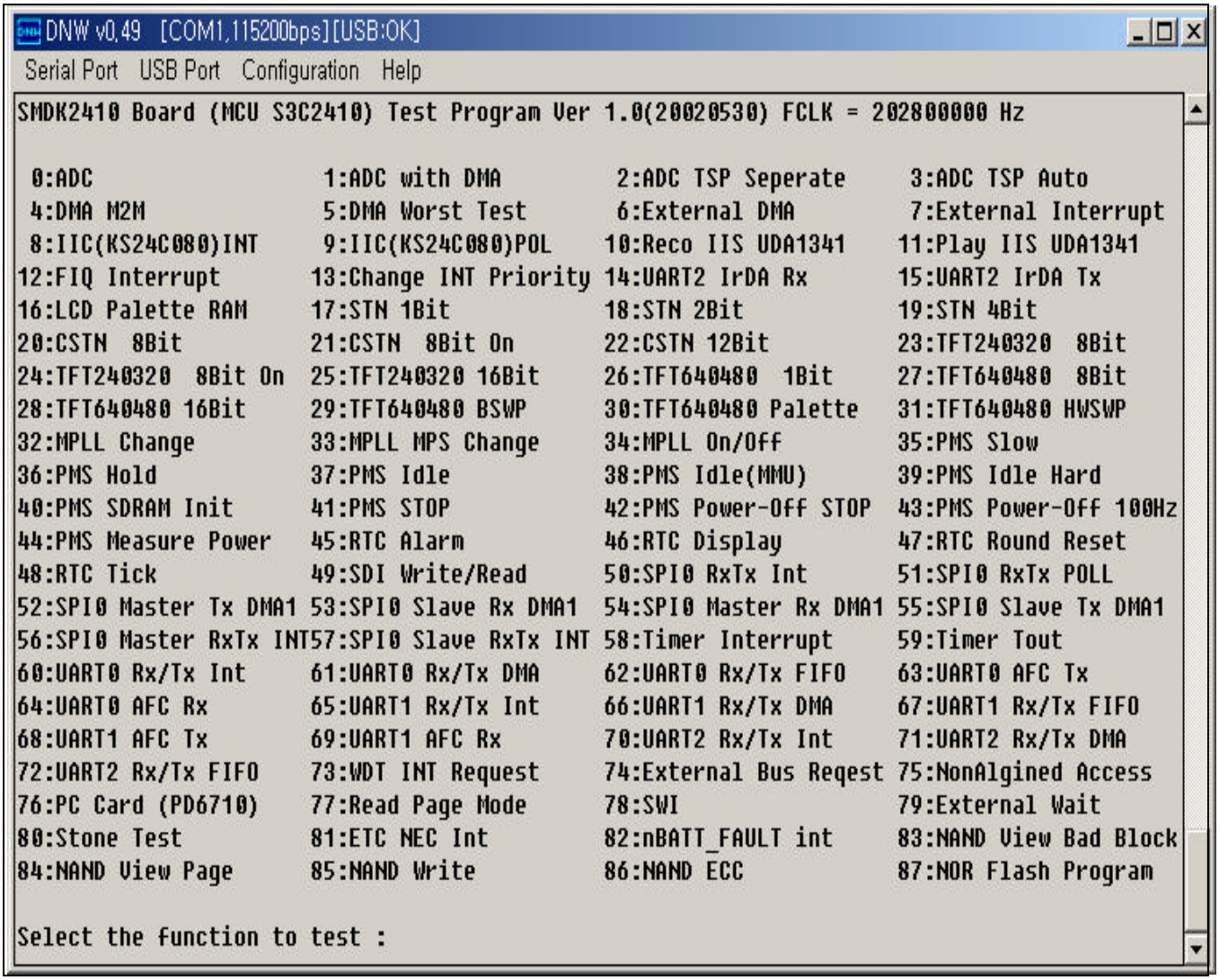


Figure 3-10. DNW Window (2410test Program)

10. Write ’85’ and press enter key on the DNW window (See Figure 3-11).
11. Write the target block number that is the start block to write and press enter key on the DNW window (See Figure 3-11).
12. Write total byte size of the target image, it must be aligned 0x4000 (one block size) bytes (See Figure 3-11).

NOTE: 2410test program supports only K9S1208 (SmartMedia card, SAMSUNG). To write another type of device, it is required to modify the source code (K9S1208.C).

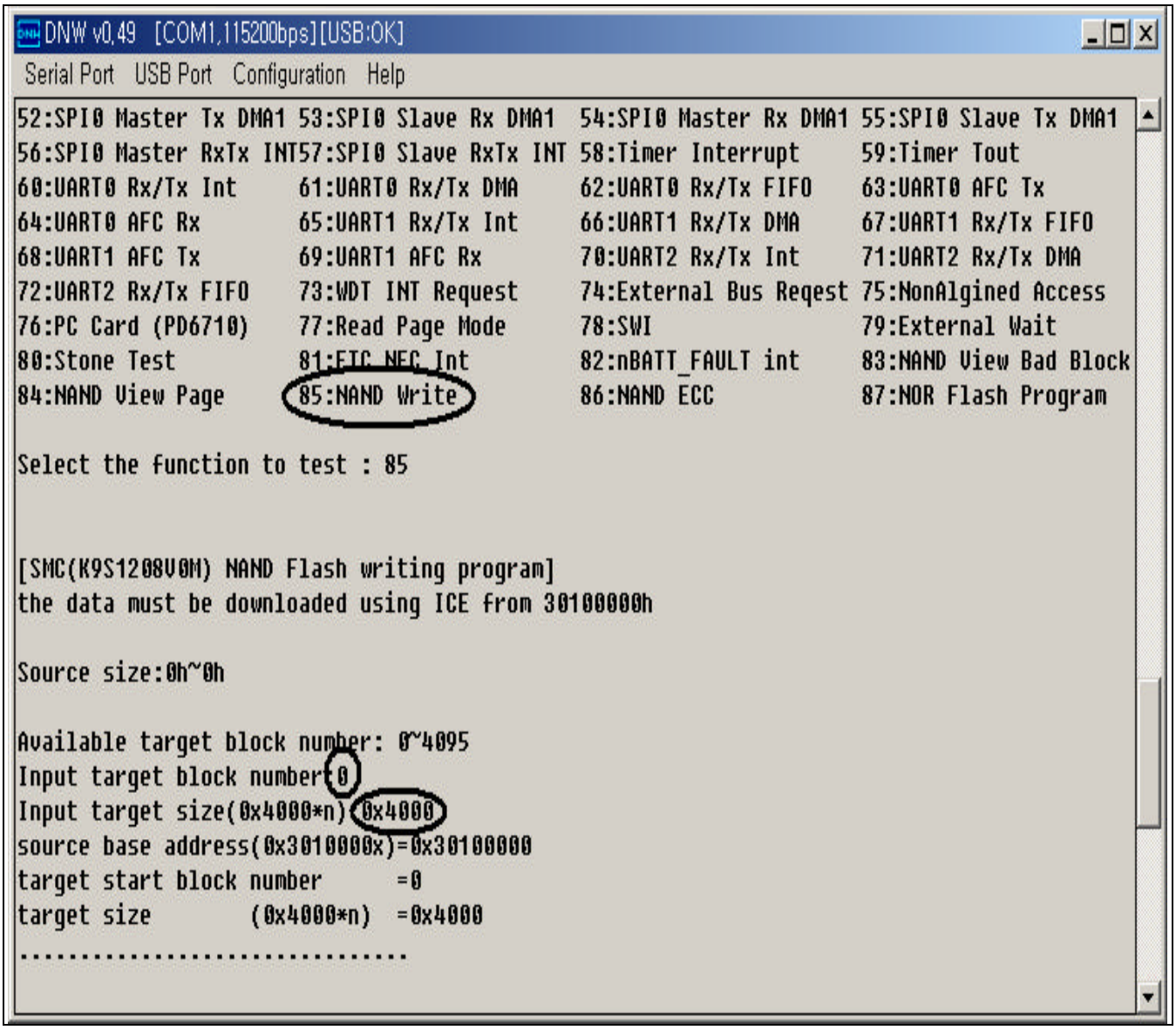


Figure 3-11. DNW Window (NAND Flash Write Program)

12. To check the contents of NAND flash, select NAND View Page function.

AUTO BOOTING THROUGH NAND FLASH

The S3C2410 supports auto booting operation with NAND flash memory.

Before power on the SMDK2410 system, it must have SmartMedia card with boot-loader and OS image.

NOTES:
1. Connector J33 must be short.
2. 2410test program and boot-loader, which is supplied by SMSUNG, support only K9S1208 (SmartMedia card, SAMSUNG). To use another type of device, it is required to modify the source code (K9S1208.C).

BOOTING NAND FLASH

To make NAND flash memory for auto booting, follow the steps:

1. Program the boot-loader image to the block 0 of NAND flash memory.
2. Program the OS image to the other blocks of NAND flash memory. The OS image must be located block 1 to the rest blocks.

NOTE: The boot-loader program, which is provided by SAMSUNG, copies 830 blocks from NAND flash memory to SDRAM.

NAND FLASH ECC (ERROR CHECKING AND CORRECTION)

The S3C2410X supports ECC algorithm, which is based on XOR calculation, for error checking and correction.

1. Example of one byte ECC (find error bit)

Old

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0

Old ECC code

P3	NP3	P2	NP2	P1	NP1
1	0	1	0	1	0

New

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	0	0	0

New ECC code

P4	NP3	P2	NP2	P1	NP1
0	0	0	0	1	1

Old ECC ^ New ECC

S3	NS3	S2	NS2	S1	NS1
1	0	1	0	0	1

P3 = [7]^[6]^[5]^[4]
P2 = [7]^[6]^[3]^[2]
P1 = [7]^[5]^[3]^[1]

NP3 = [3]^[2]^[1]^[0]
NP2 = [5]^[4]^[1]^[0]
NP1 = [6]^[4]^[2]^[0]

	P3	NP3	P2	NP2	P1	NP1
Old	1	0	1	0	1	0
New	0	0	0	0	1	1
(Old ECC) ^ (New ECC)	1	0	1	0	0	1
Error code	1		1		0	
Result	Bit 6					

NOTES:
1. [n] means bit [n] (or Dn).
2. The ’10b’ value of ‘Old ECC ^ New ECC’ corresponds to ‘1b’ Error code and ‘01b’ corresponds to ‘0b’.

2. Example of n byte ECC

	D7	D6	D5	D4	D3	D2	D1	D0
0th byte	[7]0	[6]0	[5]0	[4]0	[3]0	[2]0	[1]0	[0]0
1st byte	[7]1	[6]1	[5]1	[4]1	[3]1	[2]1	[1]1	[0]1
2nd byte	[7]2	[6]2	[5]2	[4]2	[3]2	[2]2	[1]2	[0]2
3rd byte	[7]3	[6]3	[5]3	[4]3	[3]3	[2]3	[1]3	[0]3
…	…							
509th byte	[7]509	[6]509	[5]509	[4]509	[3]509	[2]509	[1]509	[0]509
510th byte	[7]510	[6]510	[5]510	[4]510	[3]510	[2]510	[1]510	[0]510
511th byte	[7]511	[6]511	[5]511	[4]511	[3]511	[2]511	[1]511	[0]511

— Column parity (CPn)

CP3 = [7]0^ [7]1^[7]2^[7]3^ … ^[7]509^[7]510^[7]511 ^[6]0^ [6]1^[6]2^[6]3^ … ^[6]509^[6]510^[6]511 ^[5]0^ [5]1^[5]2^[5]3^ … ^[5]509^[5]510^[5]511 ^[4]0^ [4]1^[4]2^[4]3^ … ^[4]509^[4]510^[4]511	NCP3 = [3]0^ [3]1^[3]2^[3]3^ … ^[3]509^[3]510^[3]511 ^[2]0^[2]1^[2]2^[2]3^… ^[2]509^[2]510^[2]511 ^[1]0^[1]1^[1]2^[1]3^ … ^[1]509^[1]510^[1]511 ^[0]0^[0]1^[0]2^[0]3^ … ^[0]509^[0]510^[0]511

CP2 = [7]0^ [7]1^[7]2^[7]3^ … ^[7]509^[7]510^[7]511 ^[6]0^ [6]1^[6]2^[6]3^ … ^[6]509^[6]510^[6]511 ^[3]0^ [3]1^[3]2^[3]3^ … ^[3]509^[3]510^[3]511 ^[2]0^[2]1^[2]2^[2]3^… ^[2]509^[2]510^[2]511	NCP2 = [5]0^ [5]1^[5]2^[5]3^ … ^[5]509^[5]510^[5]511 ^[4]0^ [4]1^[4]2^[4]3^ … ^[4]509^[4]510^[4]511 ^[1]0^[1]1^[1]2^[1]3^ … ^[1]509^[1]510^[1]511 ^[0]0^[0]1^[0]2^[0]3^ … ^[0]509^[0]510^[0]511

CP1 = [7]0^ [7]1^[7]2^[7]3^ … ^[7]509^[7]510^[7]511 ^[5]0^ [5]1^[5]2^[5]3^ … ^[5]509^[5]510^[5]511 ^[3]0^ [3]1^[3]2^[3]3^ … ^[3]509^[3]510^[3]511 ^[1]0^[1]1^[1]2^[1]3^ … ^[1]509^[1]510^[1]511	NCP1 = [6]0^ [6]1^[6]2^[6]3^ … ^[6]509^[6]510^[6]511 ^ [4]0^ [4]1^[4]2^[4]3^ … ^[4]509^[4]510^[4]511 ^ [2]0^[2]1^[2]2^[2]3^… ^[2]509^[2]510^[2]511 ^ [0]0^[0]1^[0]2^[0]3^ … ^[0]509^[0]510^[0]511

— Row parity (RPn)

RP9 = [7]511^[6]511^[5]511^[4]511^[3]511^[2]511^[1]511^[0]511 ^… ^[7]384^[6]384^[5]384^[4]384^[3]384^[2]384^[1]384^[0]384 ^[7]383^[6]383^[5]383^[4]383^[3]383^[2]383^[1]383^[0]383 ^… ^[7]256^[6]256^[5]256^[4]256^[3]256^[2]256^[1]256^[0]256	NRP9 = [7]255^[6]255^[5]255^[4]255^[3]255^[2]255^[1]255^[0]255 ^… ^[7]128^[6]128^[5]128^[4]128^[3]128^[2]128^[1]128^[0]128 ^[7]127^[6]127^[5]127^[4]127^[3]127^[2]127^[1]127^[0]127 ^… ^[7]0^[6]0^[5]0^[4]0^[3]0^[2]0^[1]0^[0]0

RP8 = [7]511^[6]511^[5]511^[4]511^[3]511^[2]511^[1]511^[0]511 ^… ^[7]384^[6]384^[5]384^[4]384^[3]384^[2]384^[1]384^[0]384 [7]255^[6]255^[5]255^[4]255^[3]255^[2]255^[1]255^[0]255 ^… ^[7]128^[6]128^[5]128^[4]128^[3]128^[2]128^[1]128^[0]128	NRP8 = [7]383^[6]383^[5]383^[4]383^[3]383^[2]383^[1]383^[0]383 ^… ^[7]256^[6]256^[5]256^[4]256^[3]256^[2]256^[1]256^[0]256 ^[7]127^[6]127^[5]127^[4]127^[3]127^[2]127^[1]127^[0]127 ^… ^[7]0^[6]0^[5]0^[4]0^[3]0^[2]0^[1]0^[0]0

…

RP1 = [7]511^[6]511^[5]511^[4]511^[3]511^[2]511^[1]511^[0]511 ^[7]509^[6]509^[5]509^[4]509^[3]509^[2]509^[1]509^[0]509 ^[7]507^[6]507^[5]507^[4]507^[3]507^[2]507^[1]507^[0]507 ^… ^[7]1^[6]1^[5]1^[4]1^[3]1^[2]1^[1]1^[0]1	NRP1 = ^[7]510^[6]510^[5]510^[4]510^[3]510^[2]510^[1]510^[0]510 ^[7]508^[6]508^[5]508^[4]508^[3]508^[2]508^[1]508^[0]508 ^[7]506^[6]506^[5]506^[4]506^[3]506^[2]506^[1]506^[0]506 ^… ^[7]0^[6]0^[5]0^[4]0^[3]0^[2]0^[1]0^[0]0

3. Example of 4 bytes ECC (find 1bit error in the 4 bytes)

— Old data

	D7	D6	D5	D4	D3	D2	D1	D0
0th byte	0	0	0	0	0	0	0	0
1st byte	0	0	0	0	0	0	**1**	0
2nd byte	0	0	0	0	0	0	0	0
3rd byte	0	0	0	0	0	0	0	0

— New data

	D7	D6	D5	D4	D3	D2	D1	D0
0th byte	0	0	0	0	0	0	0	0
1st byte	0	0	**1**	0	0	0	**1**	0
2nd byte	0	0	0	0	0	0	0	0
3rd byte	0	0	0	0	0	0	0	0

— Column parity

	CP3	NCP3	CP2	NCP2	CP1	NCP1
Old data	0	1	0	1	1	0
New data	1	1	0	1	0	0
(Old ECC) ^ (New ECC)	1	0	0	0	1	0
Error code	1		0		1	
Result	5th bit					

— Row parity

	RP3	NRP3	RP2	NRP2
Old data	0	1	1	0
New data	0	0	0	0
(Old ECC) ^ (New ECC)	0	1	1	0
Error code	0		1	
Result	1st byte			

— Result: The 5th bit in the 1st byte is in error.

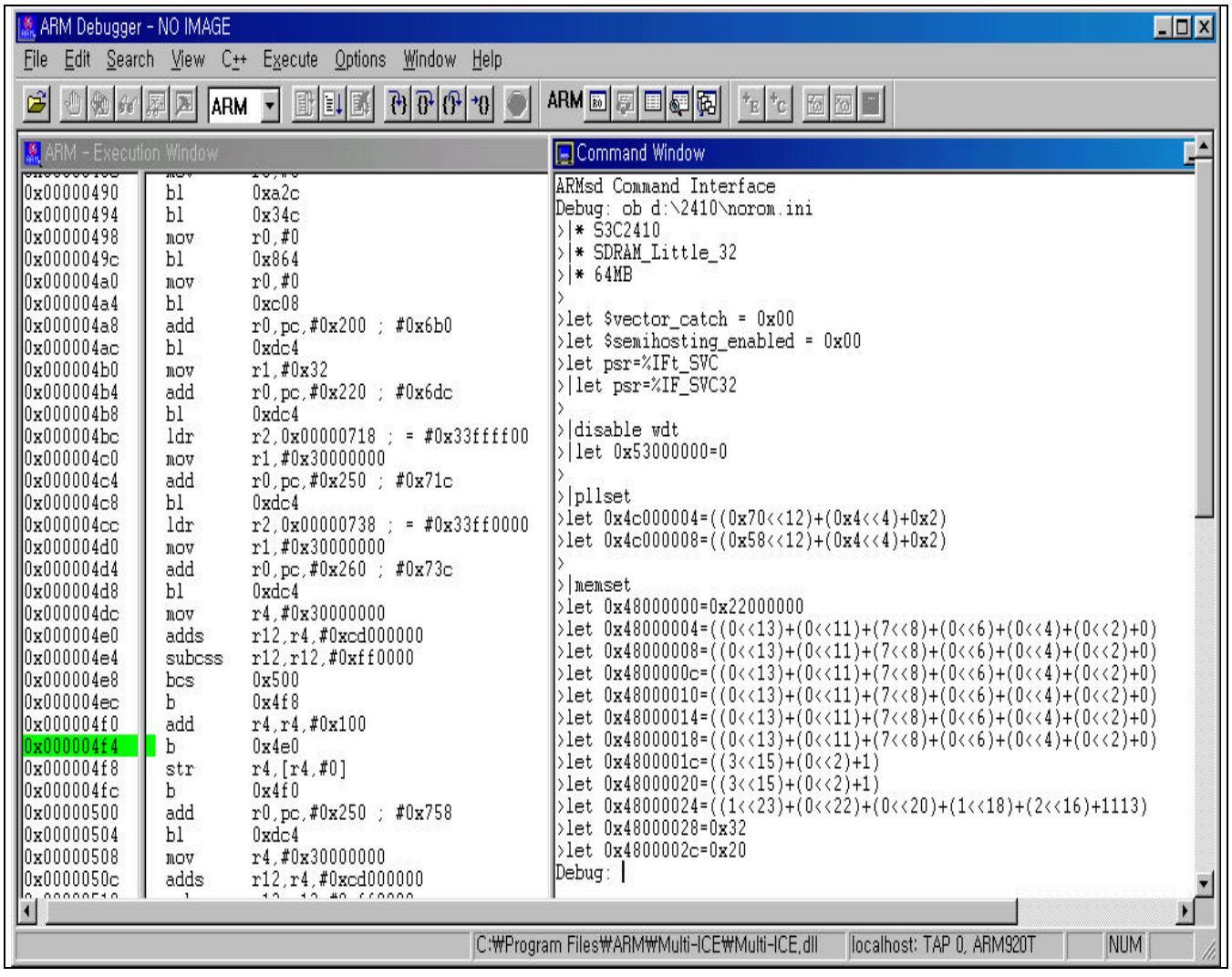
PROGRAMMING NOR FLASH MEMORY

The SMDK2410 supports NOR flash control interface. There are two types of NOR flash memories in SMDK2410: AMD and Intel STRATA flash memory. The actual methods:

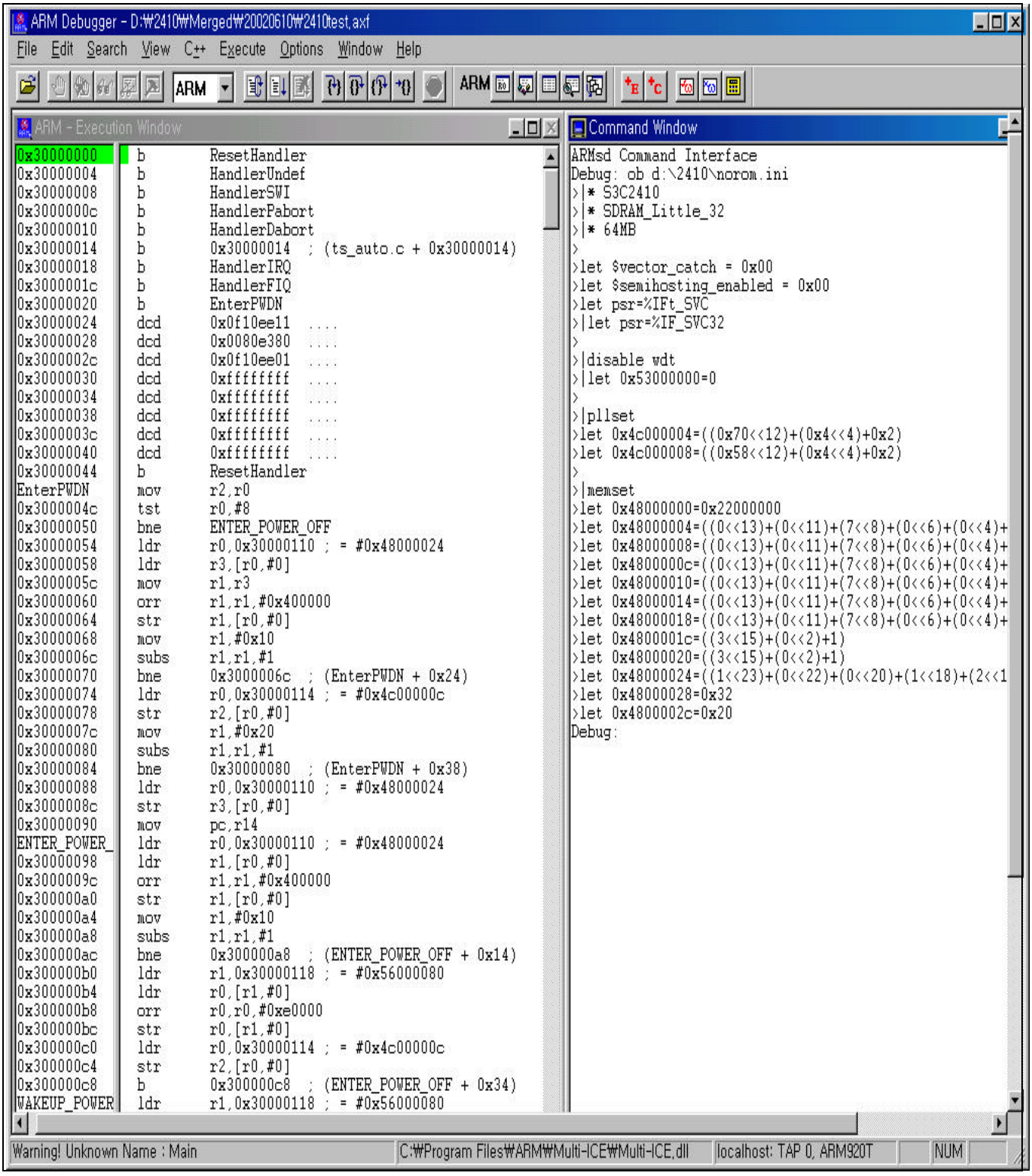
— Write image files to AMD flash memory with UART.
— Write image files to AMD flash memory with Multi-ICE.
— Write image files to AMD flash memory with OPENice32-A900
— Write image files to Intel STRATA flash memory with UART.
— Write image files to Intel STRATA flash memory with Multi-ICE.
— Write image files to Intel STRATA flash memory with OPENice32-A900

WRITING IMAGE FILES TO AMD FLASH MEMORY WITH UART

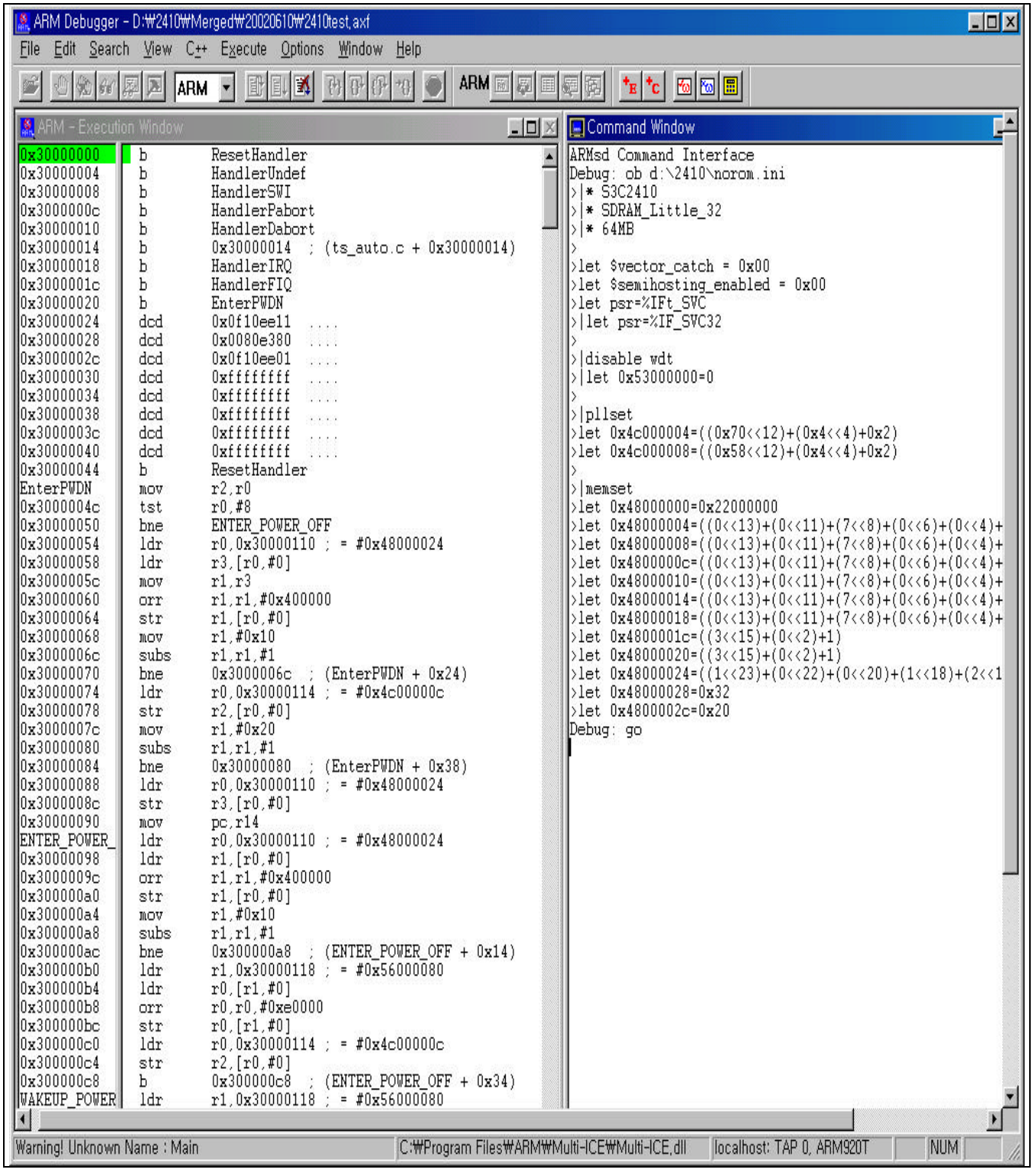
1. Connect MULTI-ICE and execute “ norom.ini “ file.



2. Load the image file (2410TEST.axf) to execute.



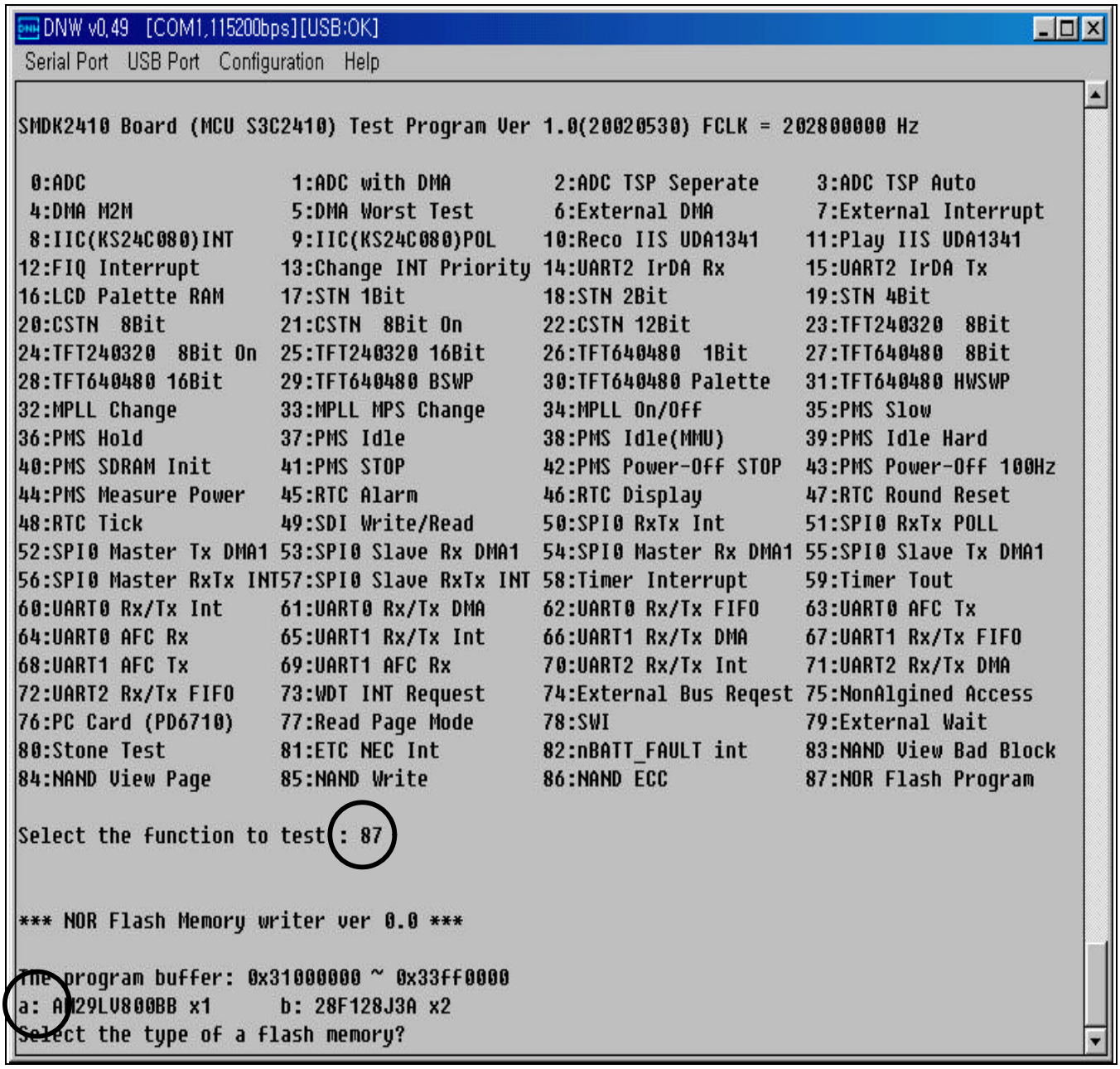
3. Execute 2410TEST code with Go command.



SAMSUNG ELECTRONICS

4. Select " 87:NOR Flash Program " on the DNW.

NOTE: If you want to download 2410TEST.bin without MULTI-ICE, then skip the 1, 2 & 3 steps above and download 2410TEST.bin using the DNW (See EXECUTE 2410TEST WITHOUT MULTI-ICE).
After download 2410TEST.bin with the DNW, then you can also see the figure below.

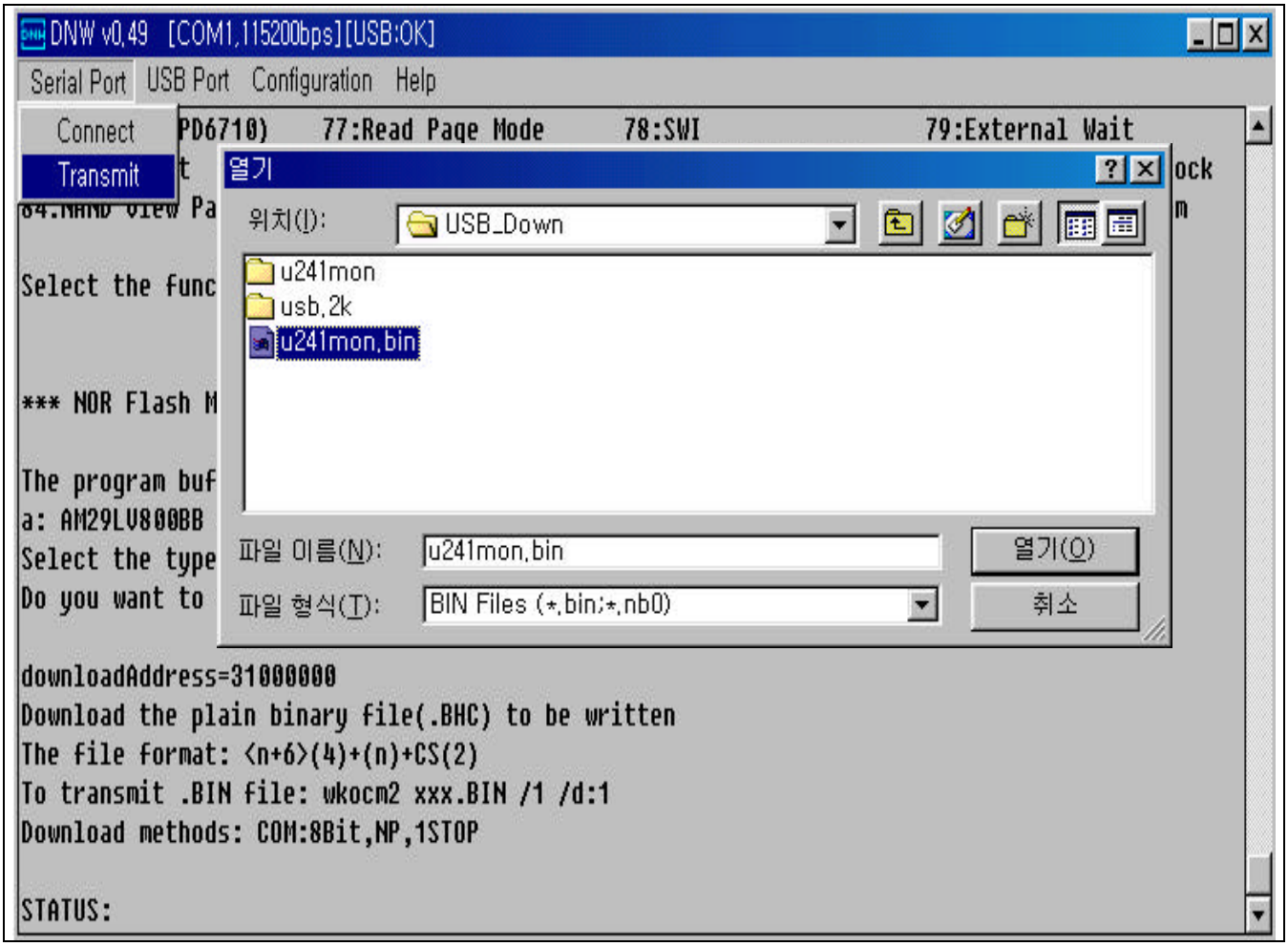


5. Select the type of memory as AM29LV800BB x1 (AMD Flash) by typing ‘a’.
6. Select whether you download through UART or MULTI-ICE.

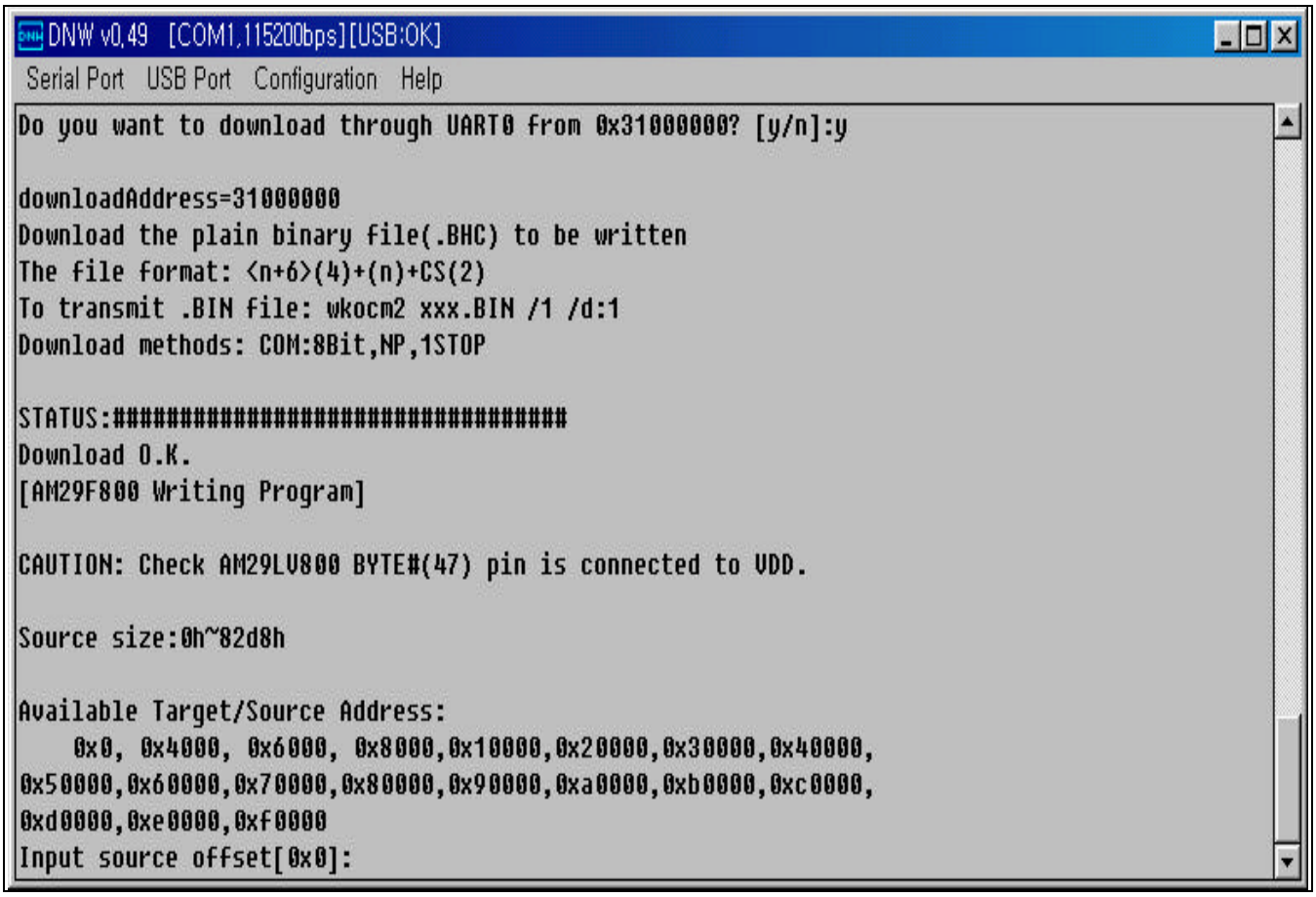
— Type ‘y’ then you can download target files through UART. See the figure below.

7. Download target files with the DNW by selecting Transmit menu from Serial Port.

— Serial Port → Transmit

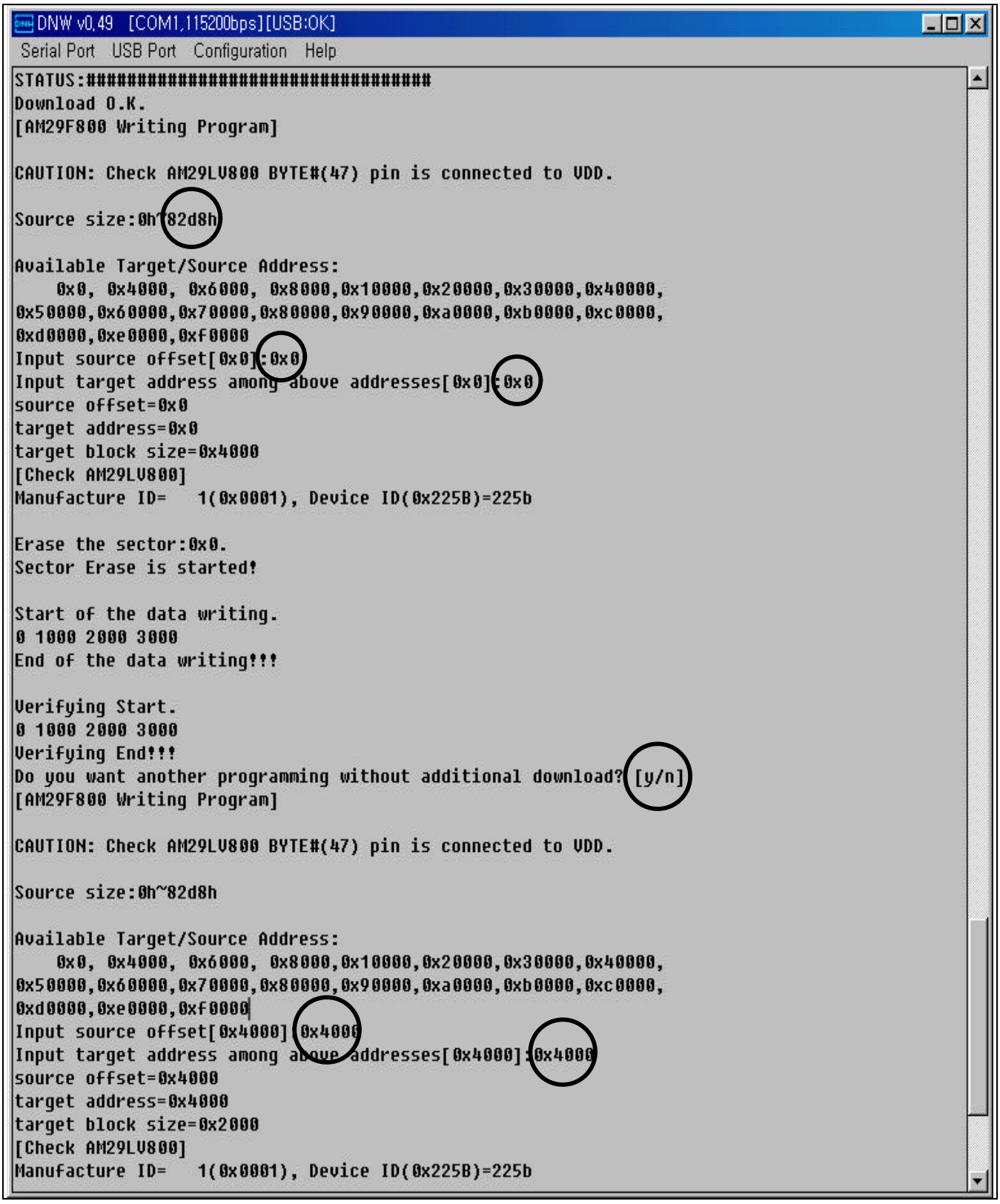


— Select and Download a target file.



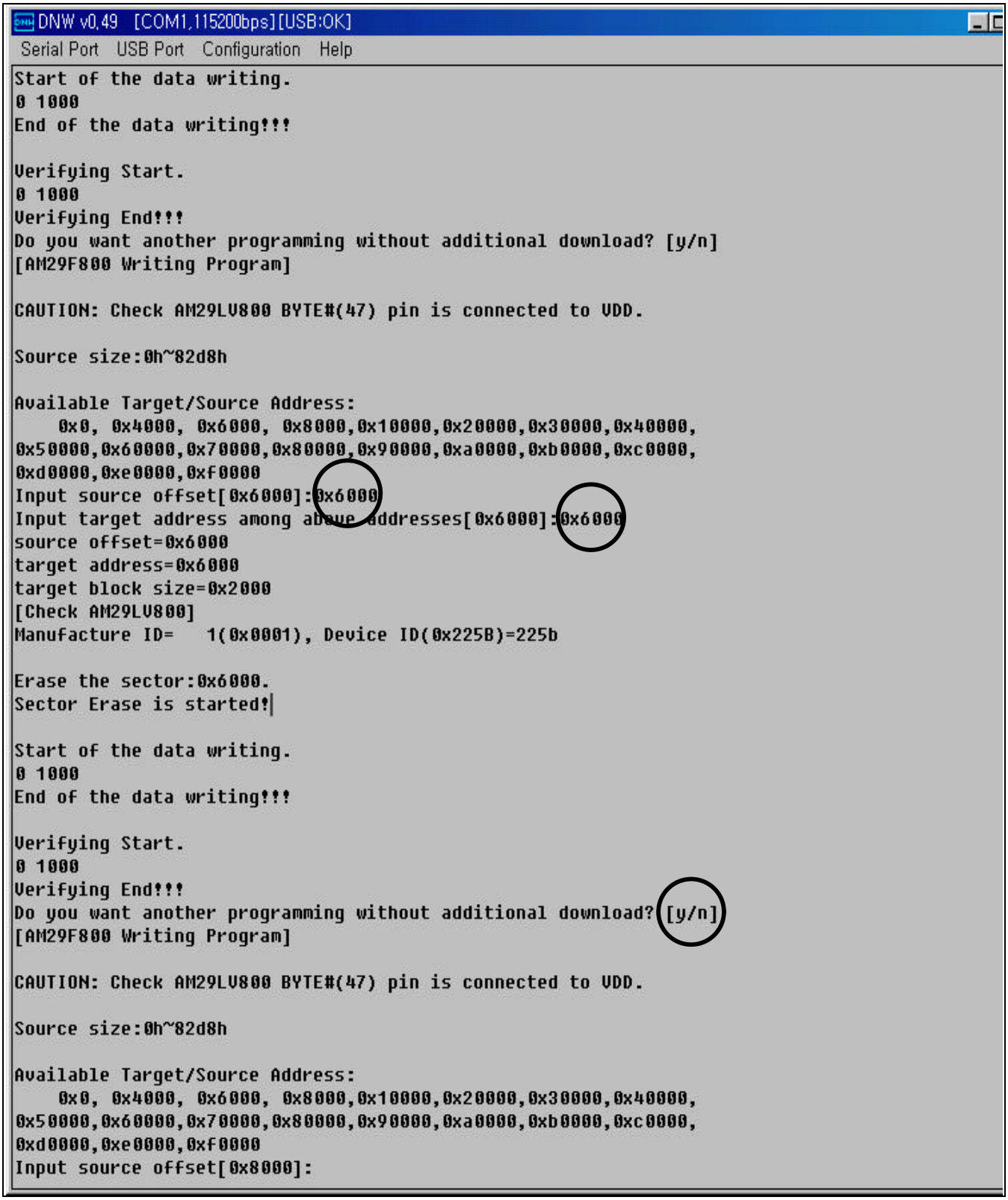
8. Write input source offset and target offset and type ‘y’ repeatedly until the source size is reached.
   The example below shows how to setting source offset and target offset. The size of target file is 0x82d8 bytes.

— Write source offset ‘ 0x0 ’ and write target address ‘ 0x0 ’, and then type ‘y’.



— Write source offset ‘ 0x4000 ’ and write target address ‘ 0x4000 ’, and then type ‘y’.

— Write source offset ‘ 0x6000 ’ and write target address ‘ 0x6000 ’, and then type ‘y’.

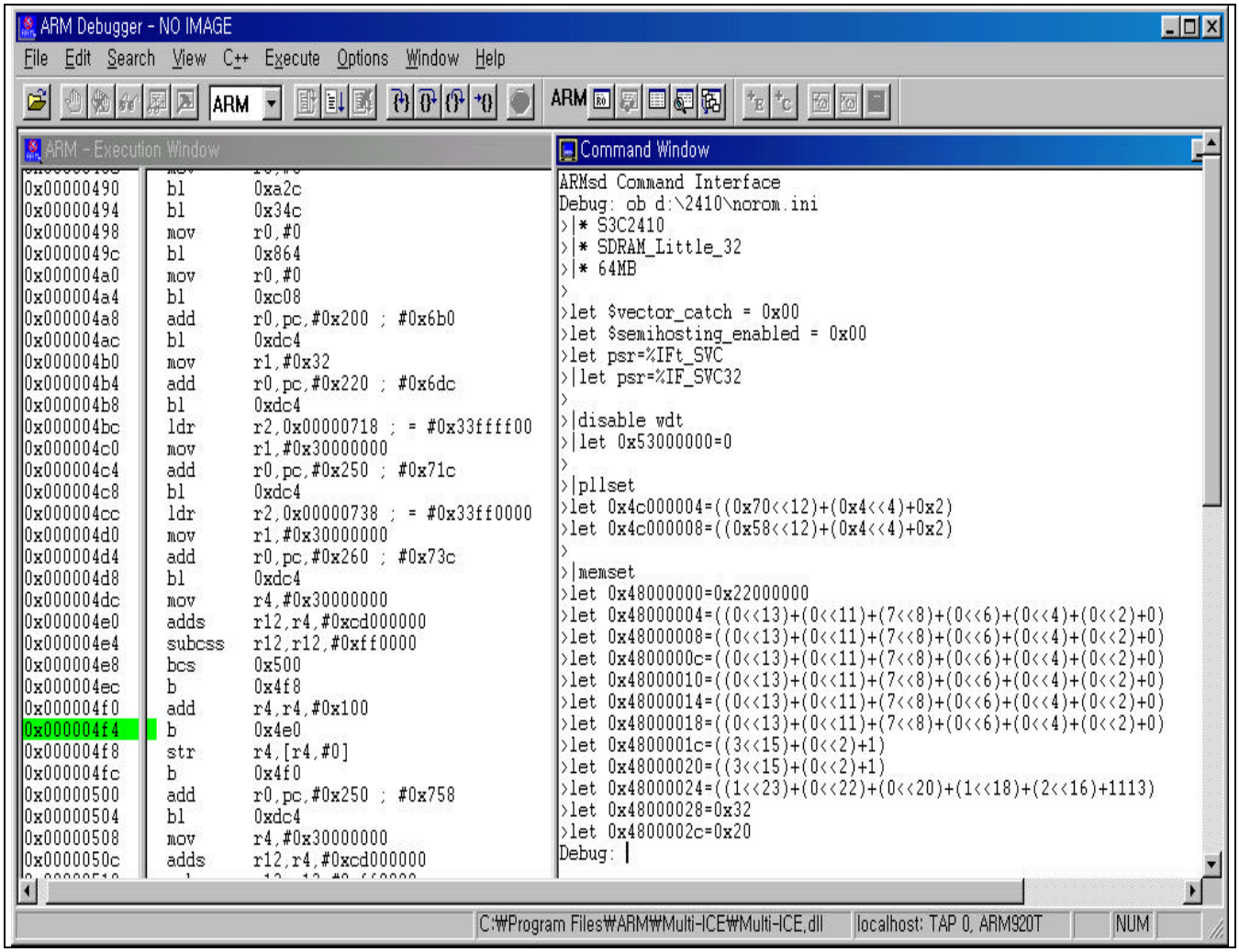


— Write source offset ‘ 0x8000 ’ and write target address ‘ 0x8000 ’, and then type ‘n’.

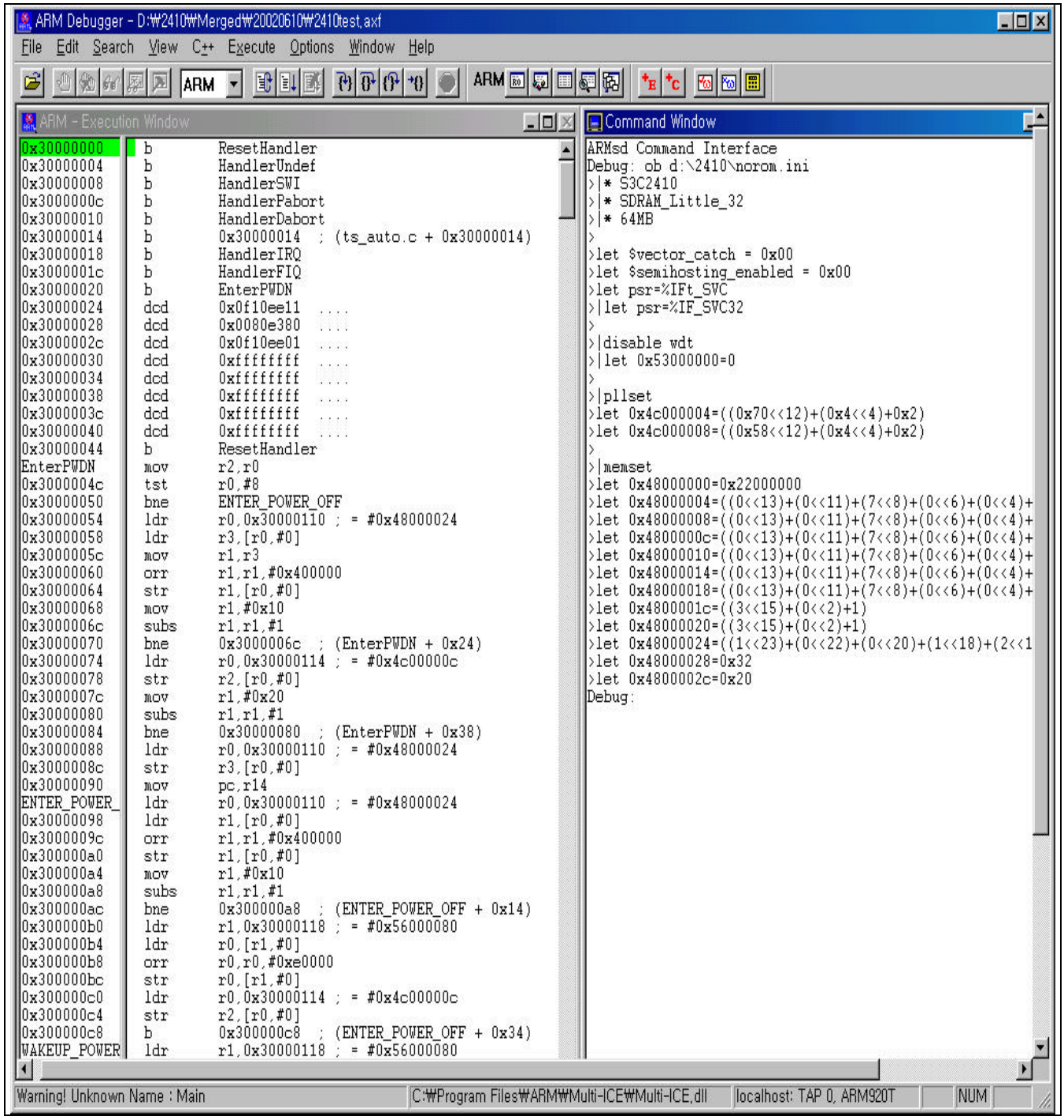
9. Turn the SMDK2410 off and then on.

WRITING IMAGE FILES TO AMD FLASH MEMORY WITH MULTI-ICE

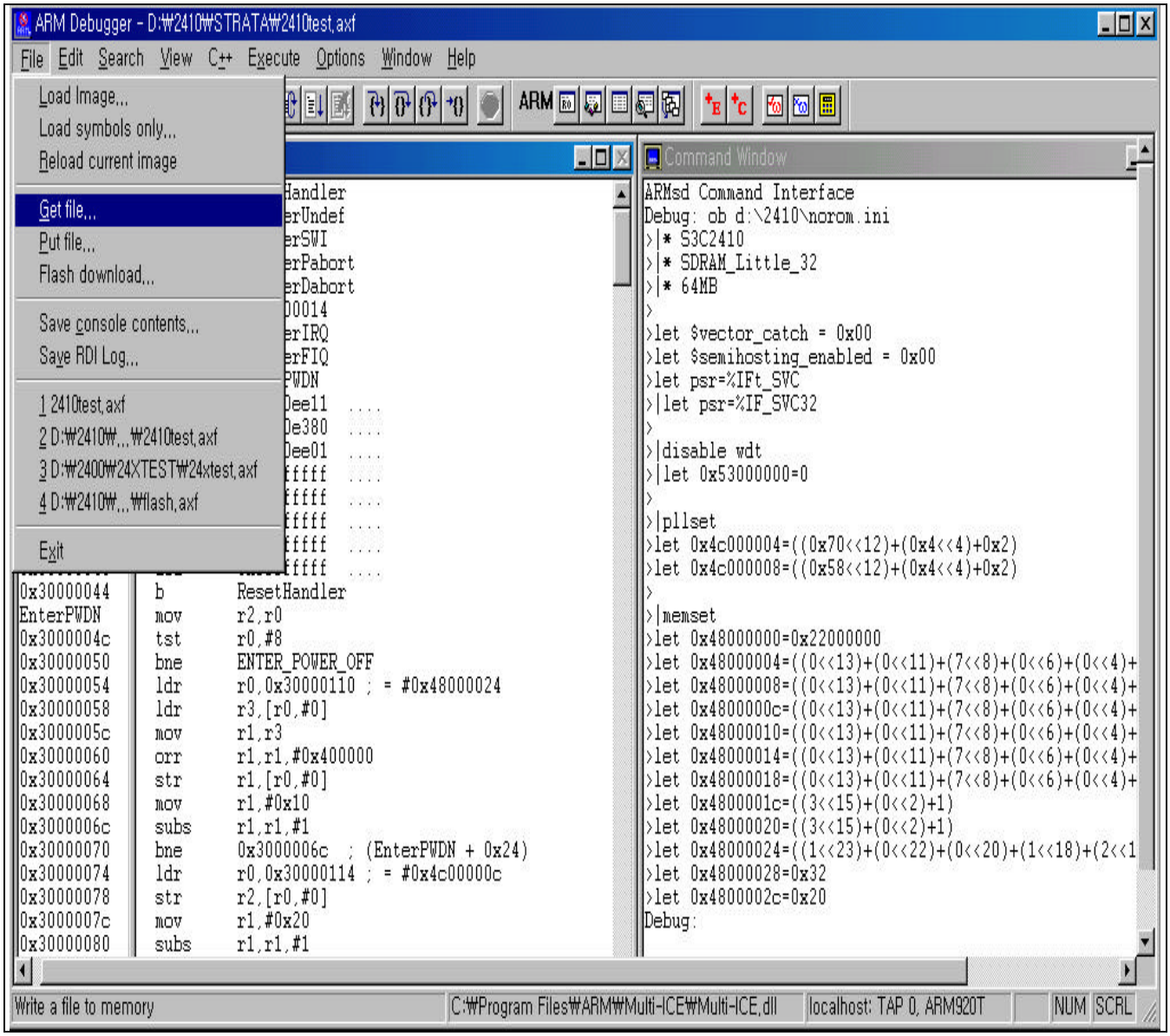
1. Connect MULTI-ICE and execute “ norom.ini “ file.



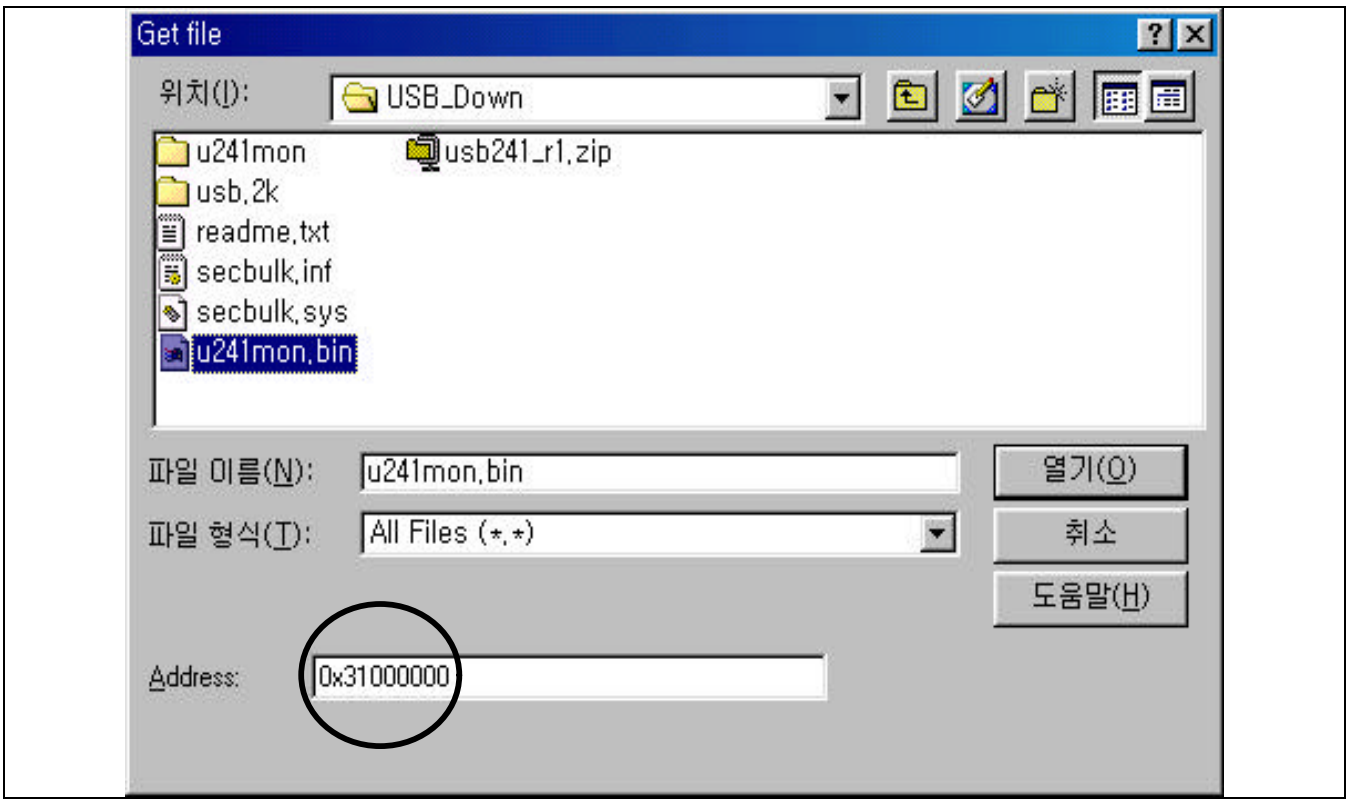
2. Load the image file (2410TEST.axf) to execute.



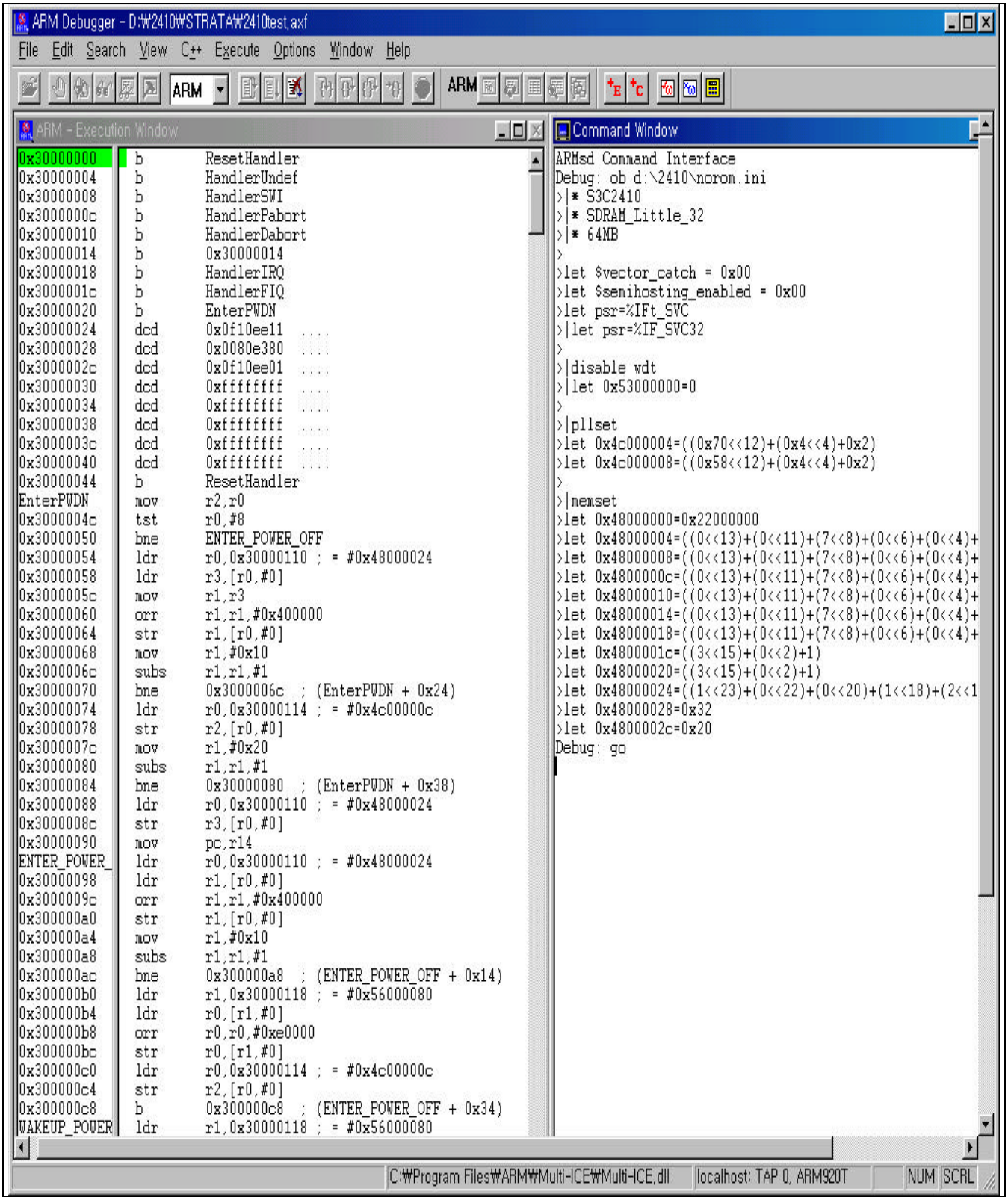
3. Get a target file written into the AMD flash memory.



— Get a target file to 0x31000000 in SMDK2410 Board.



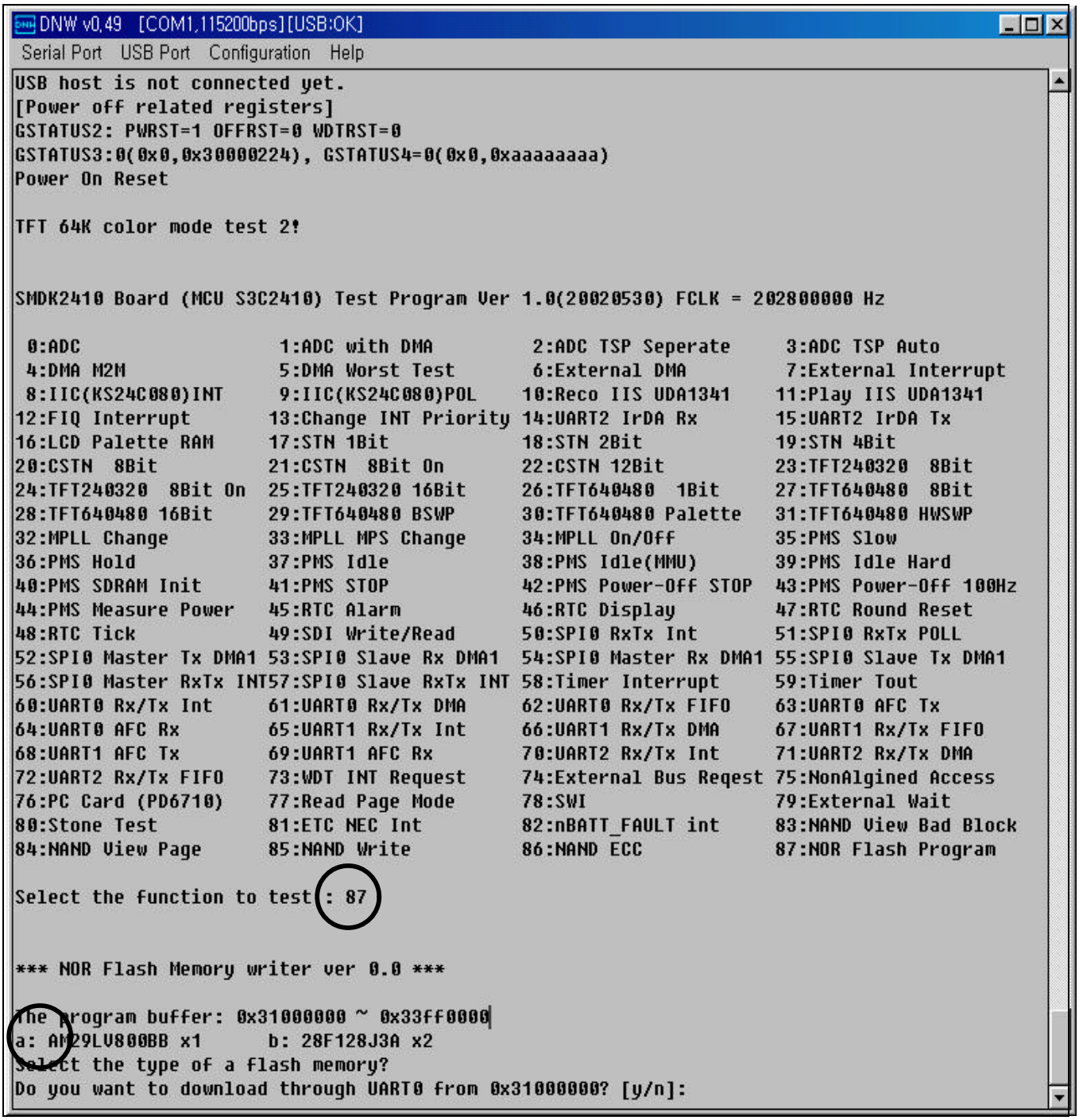
4. Execute 2410TEST.axf file with GO command.



SAMSUNG ELECTRONICS

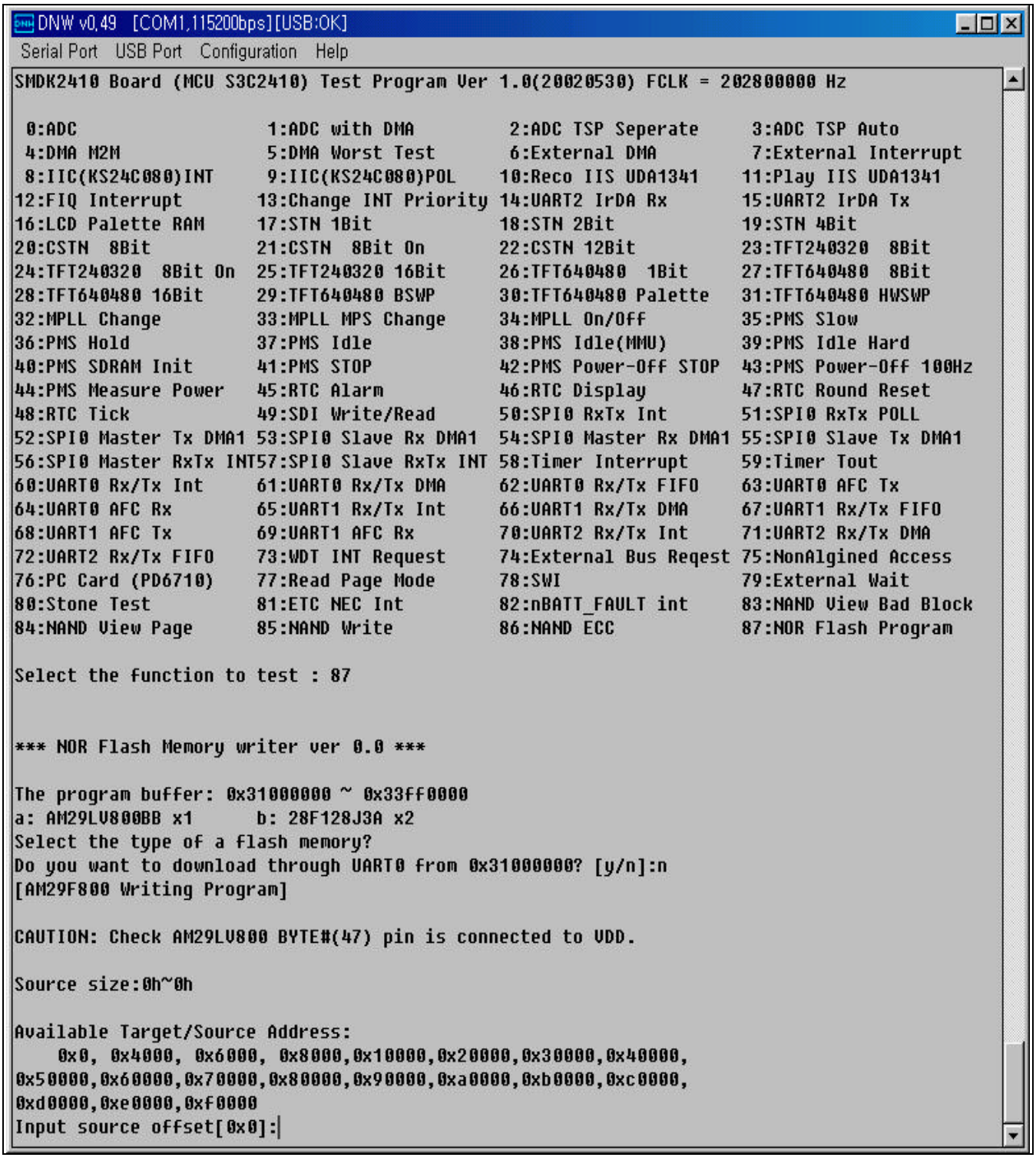
5. Select " 87:NOR Flash Program " on the DNW.

NOTE: If you want to download 2410TEST.bin without MULTI-ICE, then skip the 1, 2 & 3 steps above and download 2410TEST.bin using the DNW (See EXECUTE 2410TEST WITHOUT MULTI-ICE).
After downloading 2410TEST.bin with the DNW, then you can also see the figure below.

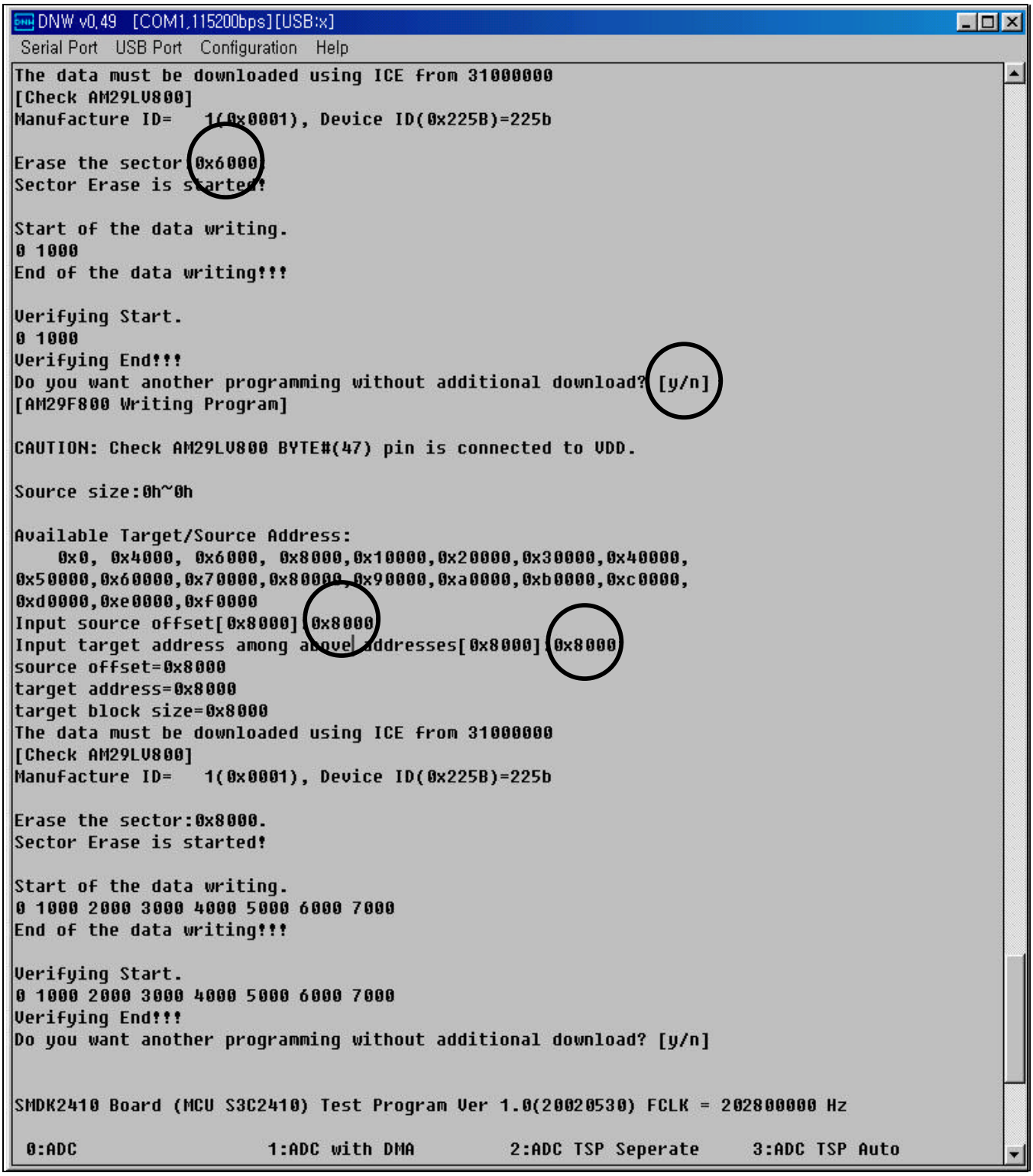


6. Select the type of memory as AM29LV800BB (AMD) by typing ‘a’.
7. Select whether you download through UART0 or MULTI-ICE.

— Type ‘n’ then you can see the figure below in the DNW.



8. Write input source offset and target offset and type ‘y’ repeatedly until the source size is reached. See Writing the image file to AMD Flash memory with UART.

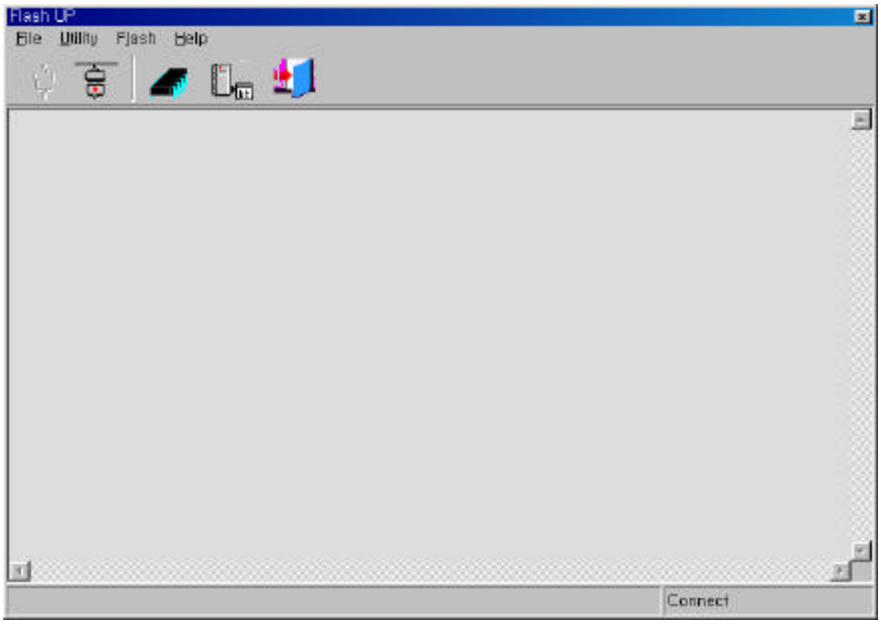


9. Turn off and on the SMDK2410.

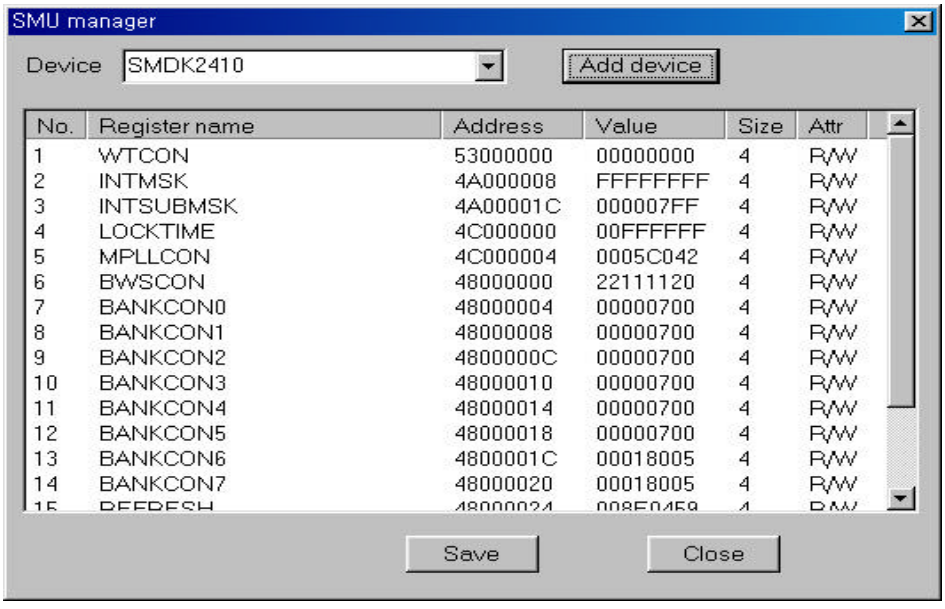
WRITING IMAGE FILES TO AMD FLASH MEMORY WITH OPENICE32-A900

OPENice32-A900 can write image to AMD Flash memory as Multi-ICE. However, OPENice32-A900 provide a Flash Write Program that is easy to use and don't require ARM SDT/ADS debugger nor DNW. For more information on the program, refer to OPENice32-A900 manual or contact AIJI System (www.aijisystem.com).

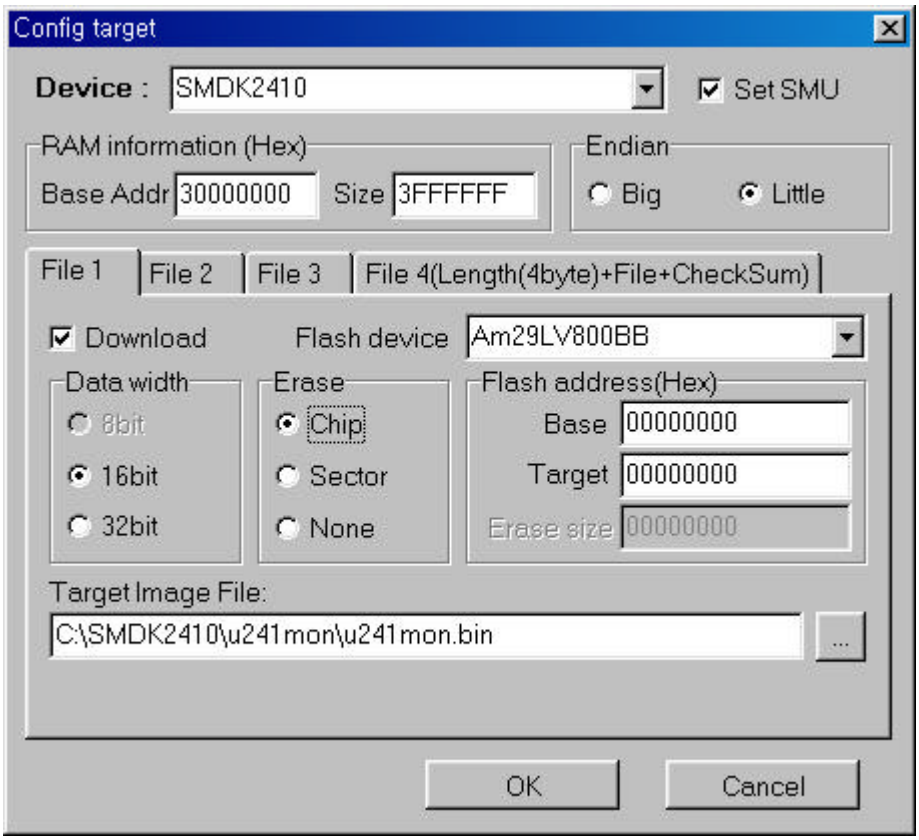
1. Connect OPENice32-A900 to PC through USB and to SMDK2410 board with 20pin Cable.
2. Run the Flash Write program and select Connect MDS from the File menu.



3. Select SMU Manger from the utility menu and choose a device file, SMDK2410. It is used to initialize the system registers in case of there is no boot ROM. If you can't find the file, download the device file from AIJI System website (www.aijisystem.com). After that, edit each value if necessary.



4. Select Config.. from the Flash menu and Set the write options as followings



— Device: SMDK2410

— Set SMU: Checked

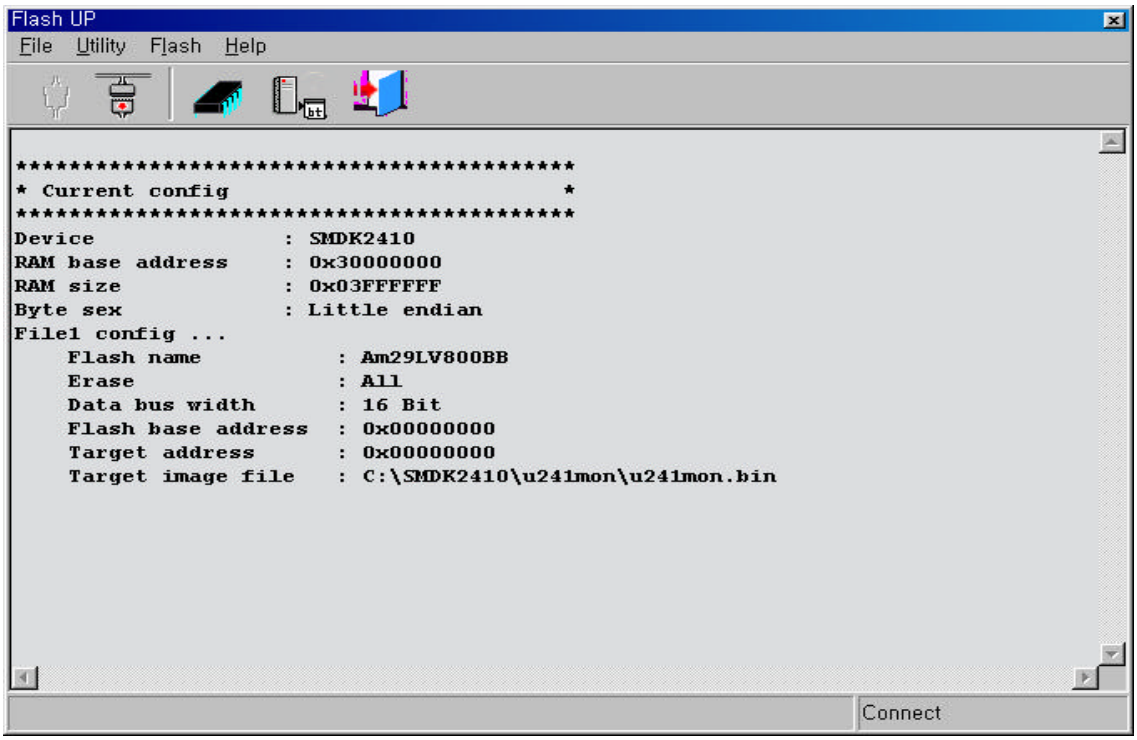
— RAM Information: Base Address:30000000 Size: 3FFFFFF

— Endian: Little

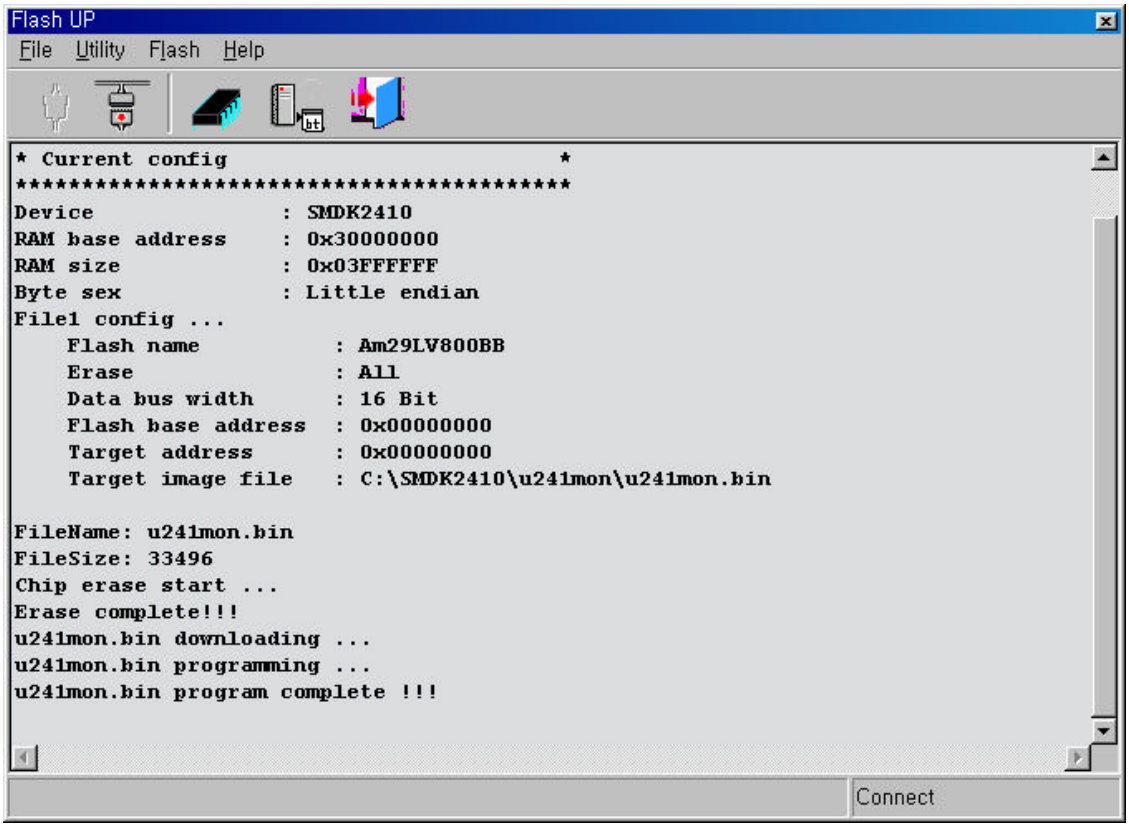
— File 1 page
Download: checked
Flash Device Name:AM29LV800BB
Erase:Chip
Data Bus width: 16bit
Flash Address: Base Address: 0 Target Address:0

— Target Image File: u241mon.bin

5. Click OK. Then the current configuration is displayed in the window.

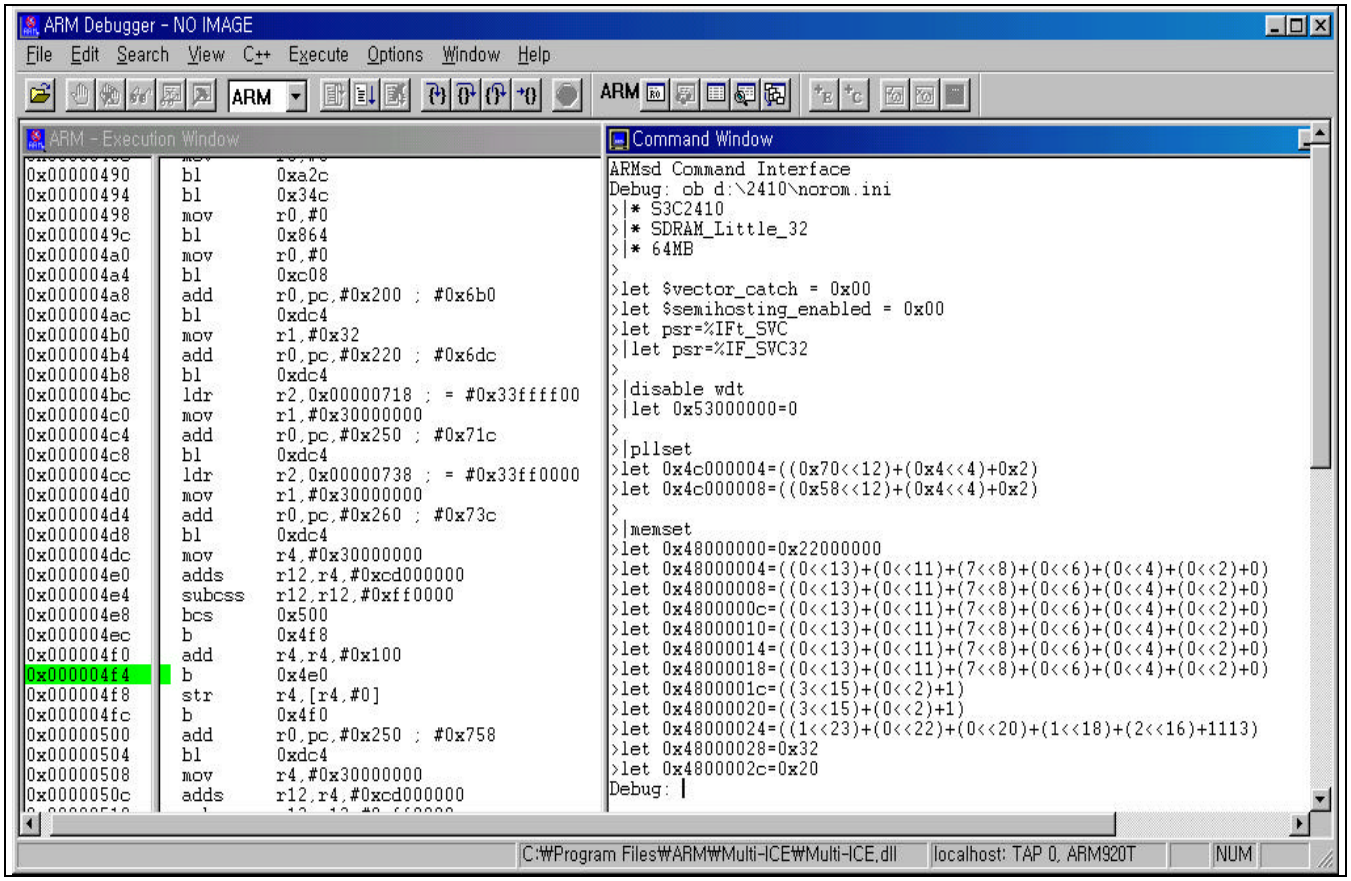


6. Select Write from the Flash Menu. Then it starts to erase the specified area of AMD Flash and write the image to the Flash memory. It takes about 10 second.



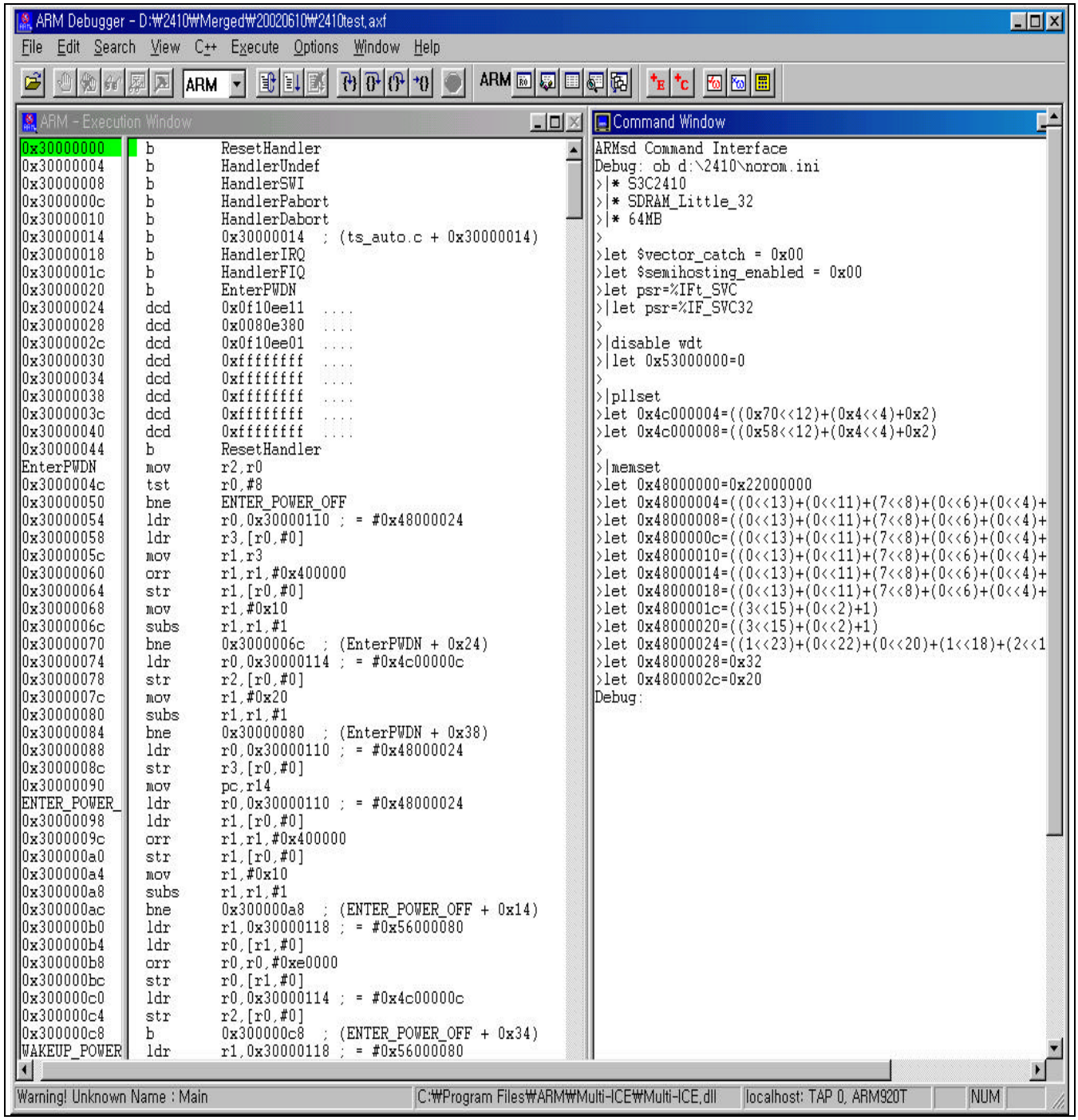
WRITING IMAGE FILES TO INTEL STRATA FLASH MEMORY WITH UART

1. Connect MULTI-ICE and execute “ norom.ini “ file.

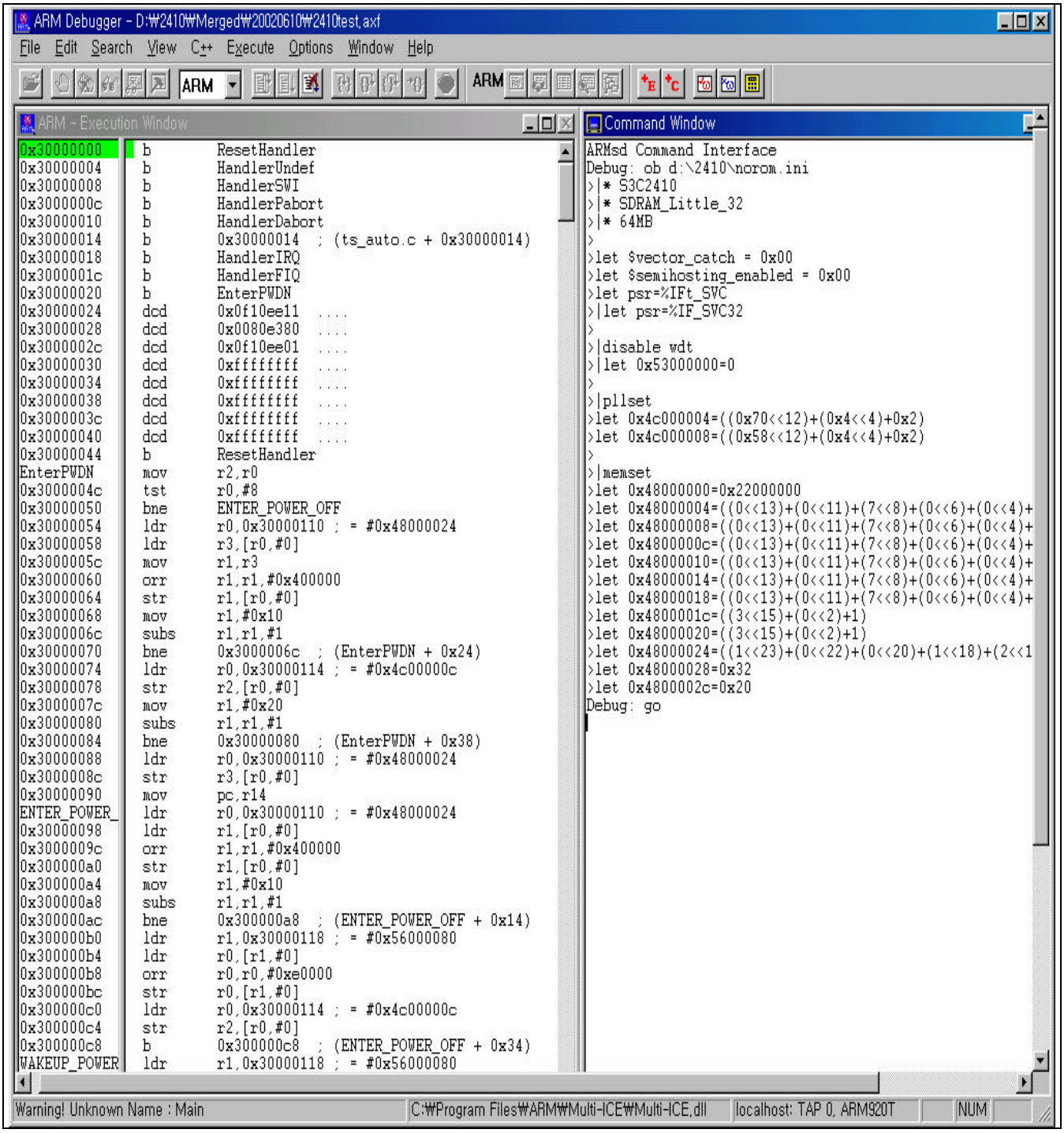


SAMSUNG ELECTRONICS

2. Load the image file (2410TEST.axf) to execute.



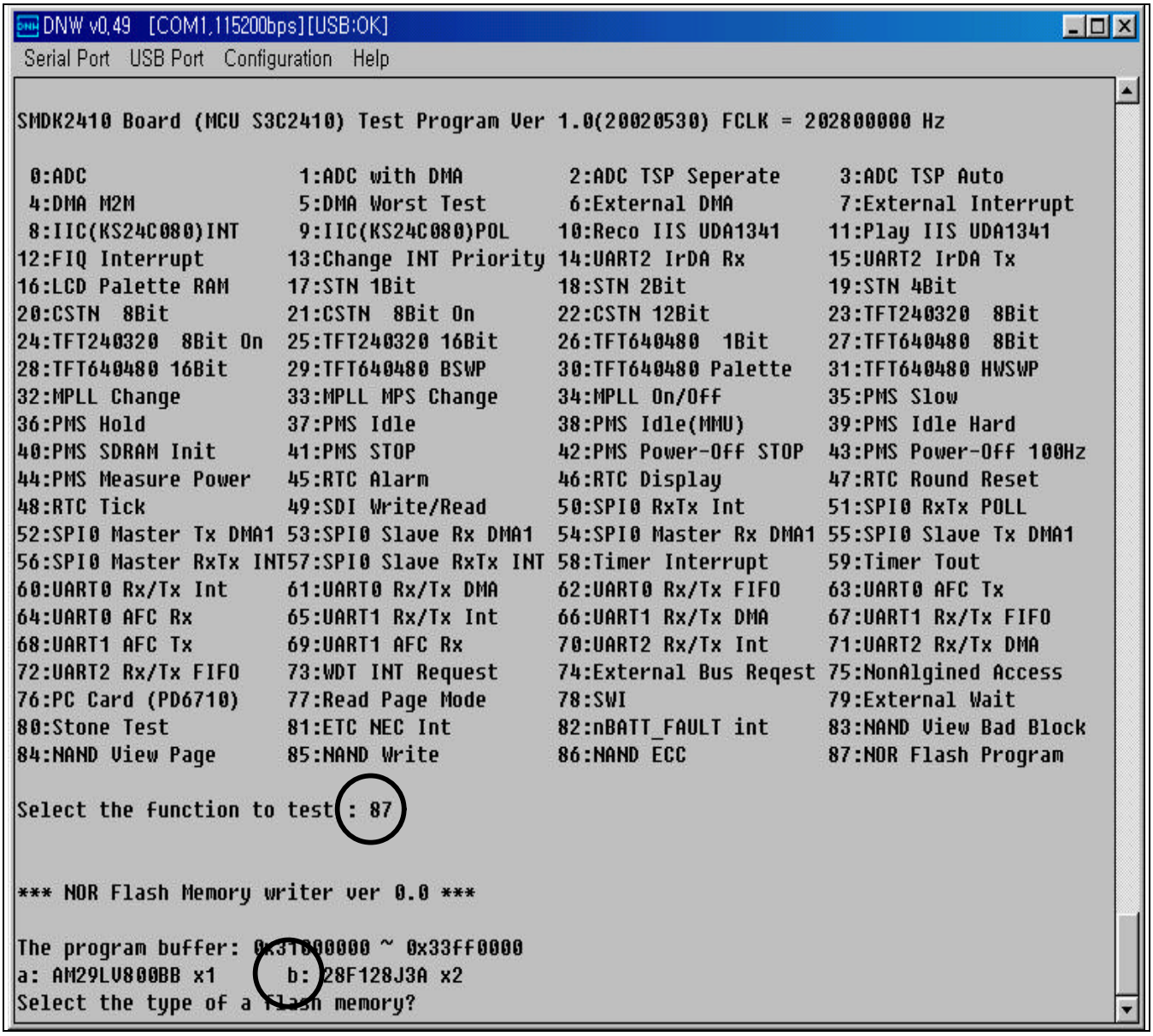
3. Execute 2410TEST code with Go command.



SAMSUNG ELECTRONICS

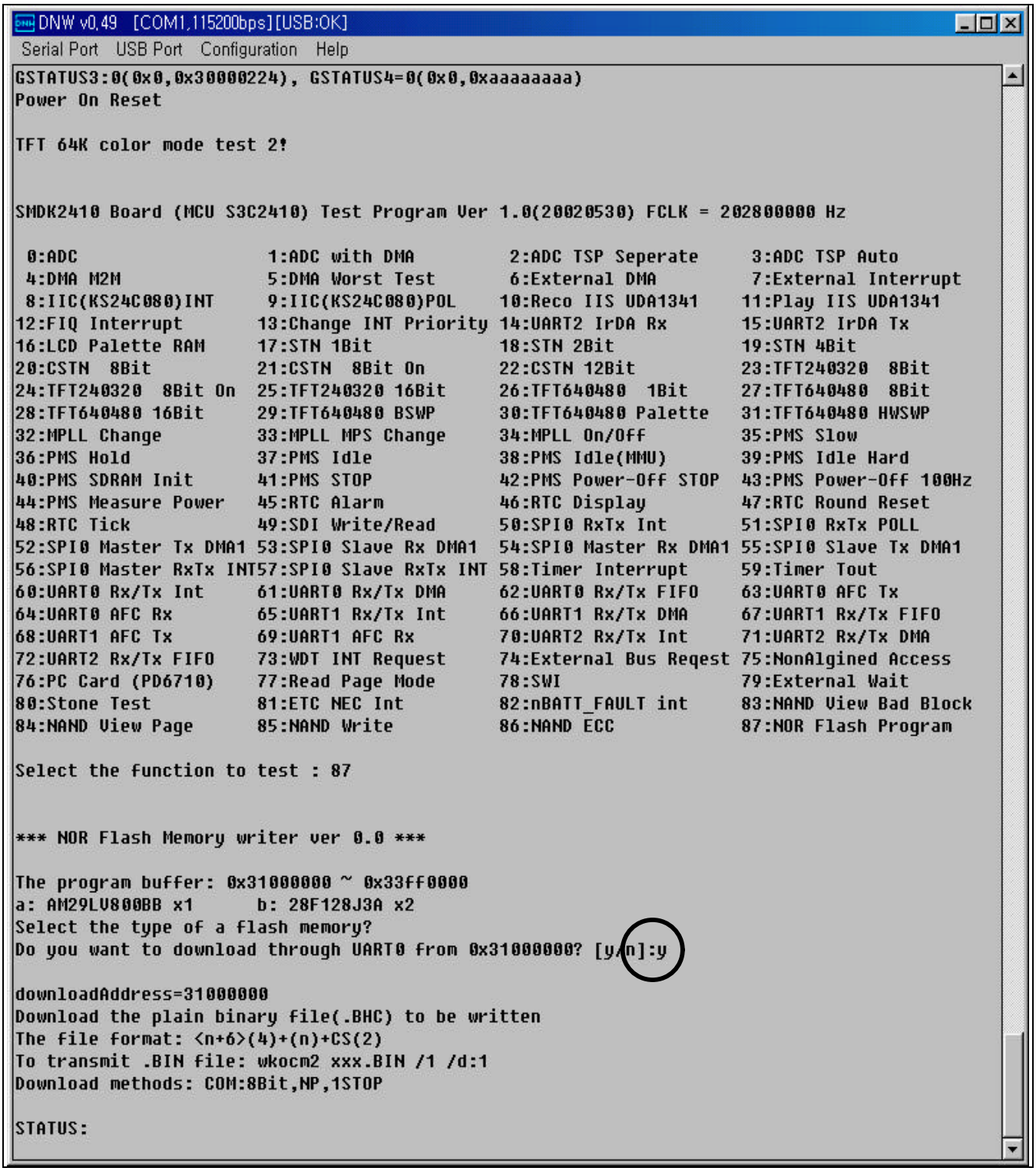
4. Select " 87:NOR Flash Program " on the DNW.

NOTE: If you want to download 2410TEST.bin without MULTI-ICE, then skip the 1, 2 & 3 steps above and download 2410TEST.bin using the DNW (See EXECUTE 2410TEST WITHOUT MULTI-ICE).
After downloading 2410TEST.bin with the DNW, then you can also see the figure below.



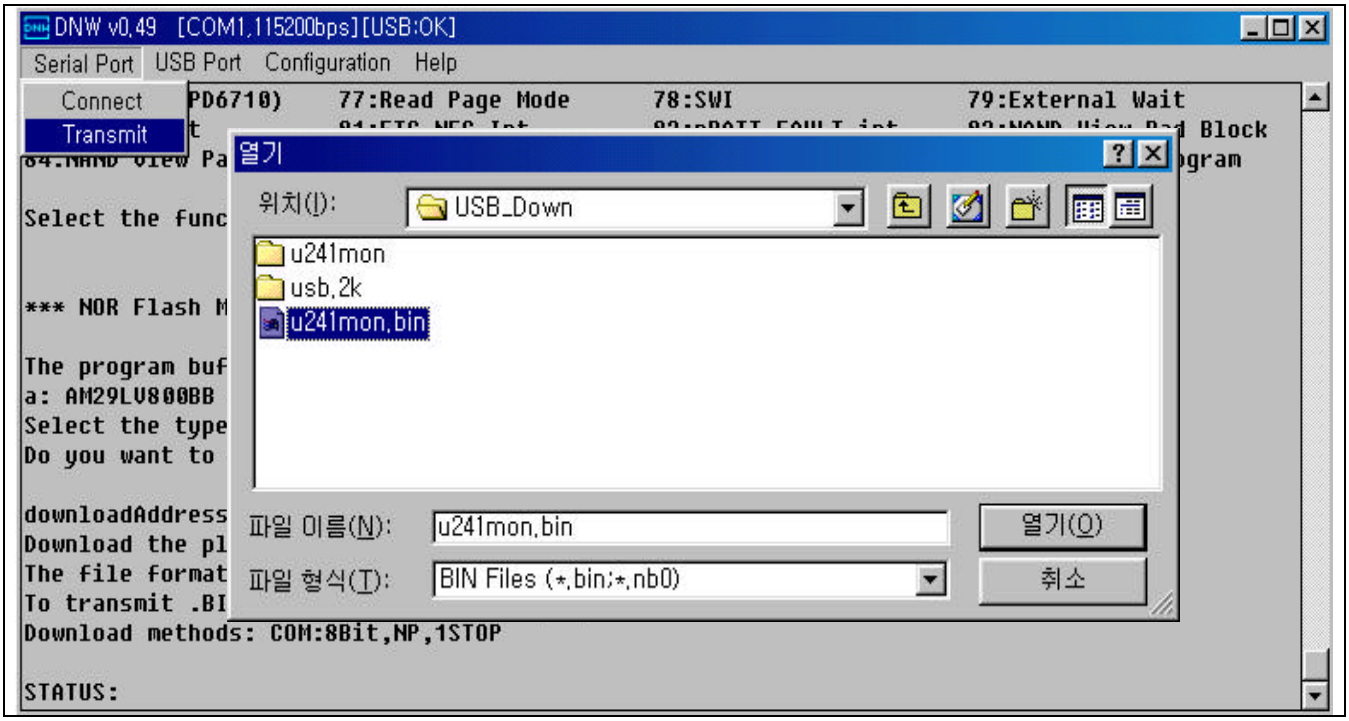
5. Select the type of memory as 28F128J3A (INTEL STRATA flash) by typing ‘b’.
6. Select whether you download through UART0 or MULTI-ICE.

— Type ‘y’ then you can download a target file through UART. See the figure below.

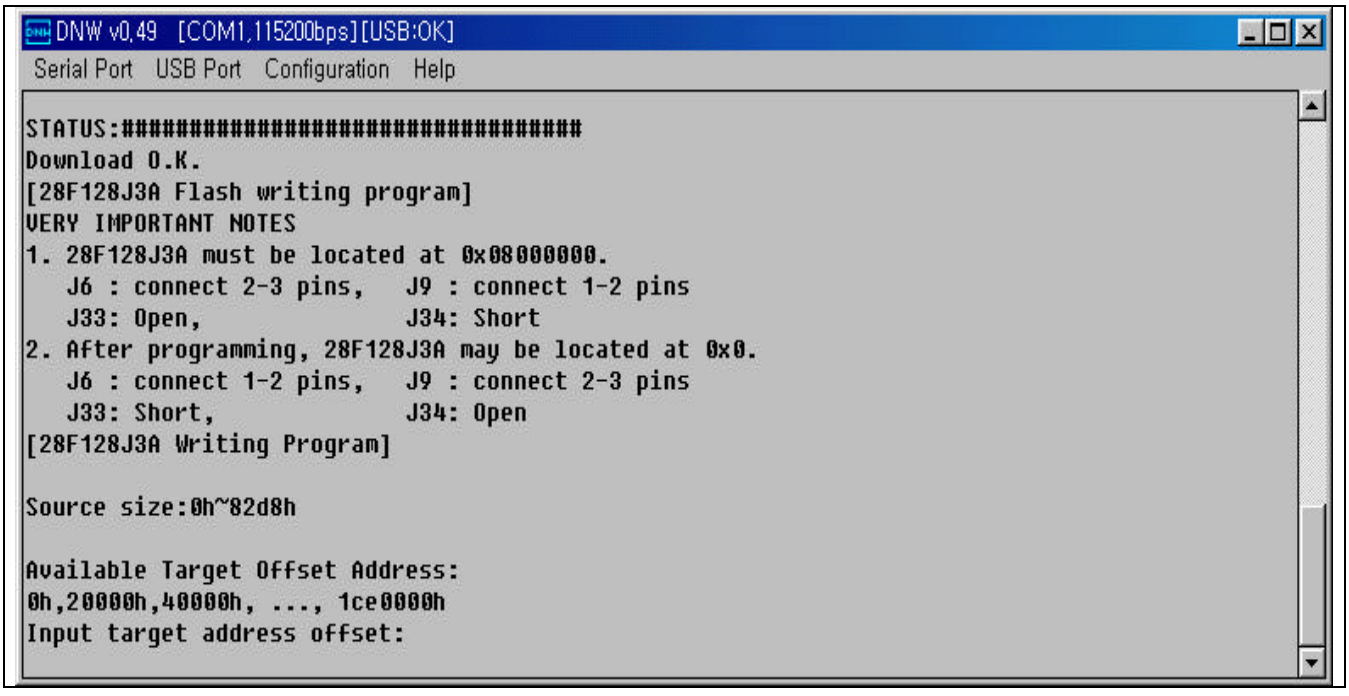

DNW v0.49 [COM1,115200bps][USB:OK]
Serial Port USB Port Configuration Help
GSTATUS3:0(0x0,0x30000224), GSTATUS4=0(0x0,0xaaaaaaaa)
Power On Reset
TFT 64K color mode test 2!
SMDK2410 Board (MCU S3C2410) Test Program Ver 1.0(20020530) FCLK = 202800000 Hz
0:ADC 1:ADC with DMA 2:ADC TSP Seperate 3:ADC TSP Auto
4:DMA M2M 5:DMA Worst Test 6:External DMA 7:External Interrupt
8:IIC(KS24C080)INT 9:IIC(KS24C080)POL 10:Reco IIS UDA1341 11:Play IIS UDA1341
12:FIQ Interrupt 13:Change INT Priority 14:UART2 IrDA Rx 15:UART2 IrDA Tx
16:LCD Palette RAM 17:STN 1Bit 18:STN 2Bit 19:STN 4Bit
20:CSTN 8Bit 21:CSTN 8Bit On 22:CSTN 12Bit 23:TFT240320 8Bit
24:TFT240320 8Bit On 25:TFT240320 16Bit 26:TFT640480 1Bit 27:TFT640480 8Bit
28:TFT640480 16Bit 29:TFT640480 BSWP 30:TFT640480 Palette 31:TFT640480 HWSWP
32:MPLL Change 33:MPLL MPS Change 34:MPLL On/Off 35:PMS Slow
36:PMS Hold 37:PMS Idle 38:PMS Idle(MMU) 39:PMS Idle Hard
40:PMS SDRAM Init 41:PMS STOP 42:PMS Power-Off STOP 43:PMS Power-Off 100Hz
44:PMS Measure Power 45:RTC Alarm 46:RTC Display 47:RTC Round Reset
48:RTC Tick 49:SDI Write/Read 50:SPI0 RxTx Int 51:SPI0 RxTx POLL
52:SPI0 Master Tx DMA1 53:SPI0 Slave Rx DMA1 54:SPI0 Master Rx DMA1 55:SPI0 Slave Tx DMA1
56:SPI0 Master RxTx INT 57:SPI0 Slave RxTx INT 58:Timer Interrupt 59:Timer Tout
60:UART0 Rx/Tx Int 61:UART0 Rx/Tx DMA 62:UART0 Rx/Tx FIFO 63:UART0 AFC Tx
64:UART0 AFC Rx 65:UART1 Rx/Tx Int 66:UART1 Rx/Tx DMA 67:UART1 Rx/Tx FIFO
68:UART1 AFC Tx 69:UART1 AFC Rx 70:UART2 Rx/Tx Int 71:UART2 Rx/Tx DMA
72:UART2 Rx/Tx FIFO 73:WDT INT Request 74:External Bus Reqest 75:NonAlgined Access
76:PC Card (PD6710) 77:Read Page Mode 78:SWI 79:External Wait
80:Stone Test 81:ETC NEC Int 82:nBATT_FAULT int 83:NAND View Bad Block
84:NAND View Page 85:NAND Write 86:NAND ECC 87:NOR Flash Program
Select the function to test : 87
*** NOR Flash Memory writer ver 0.0 ***
The program buffer: 0x31000000 ~ 0x33ff0000
a: AM29LV800BB x1 b: 28F128J3A x2
Select the type of a flash memory?
Do you want to download through UART0 from 0x31000000? [y/n]:y
downloadAddress=31000000
Download the plain binary file(.BHC) to be written
The file format: <n+6>(4)+(n)+CS(2)
To transmit .BIN file: wkocm2 xxx.BIN /1 /d:1
Download methods: COM:8Bit,NP,1STOP
STATUS:

7. Download a target file with the DNW by selecting Transmit menu from Serial Port.

— Serial Port → Transmit



— Select and Download a target file.

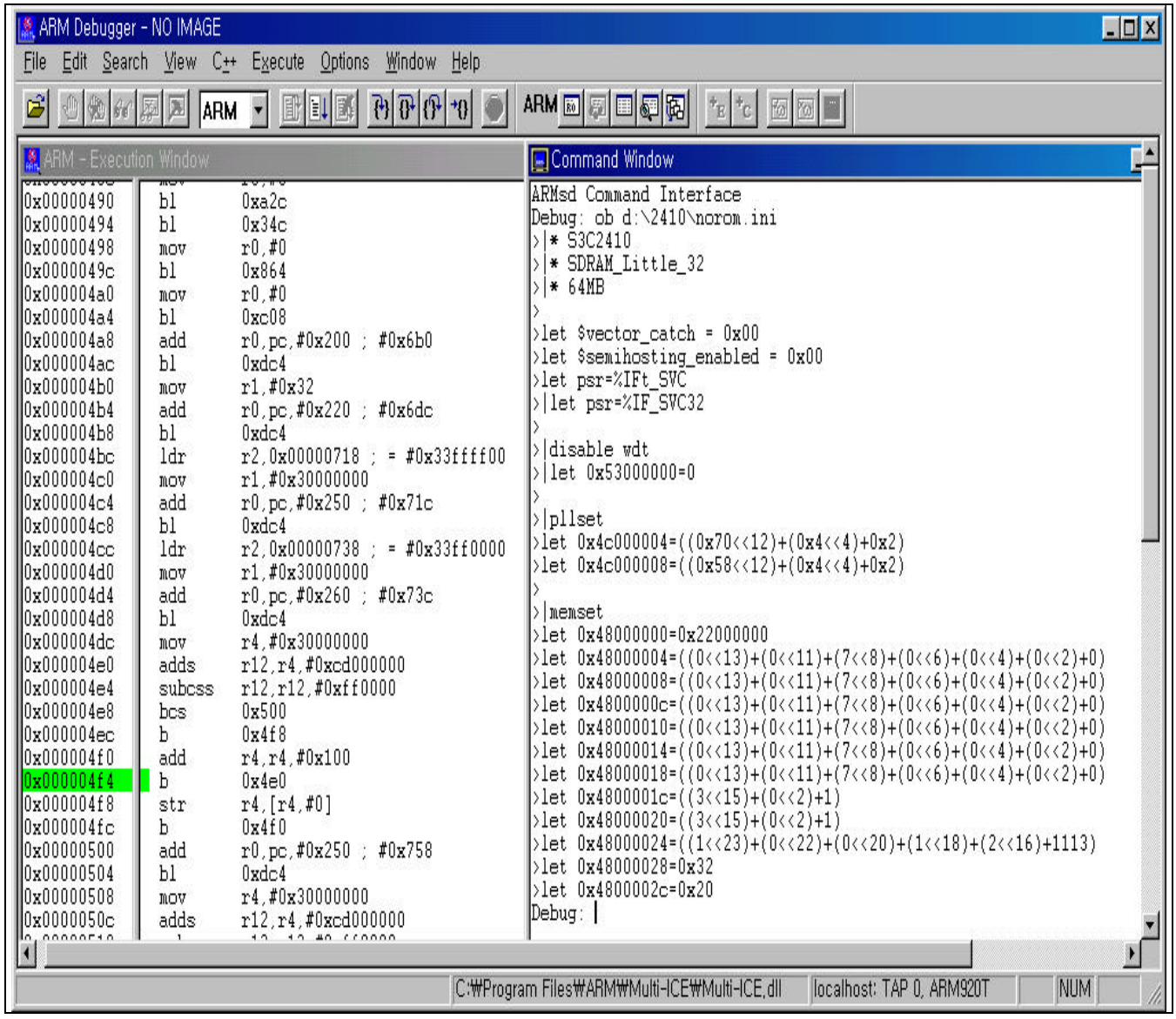


8. Write input target-offset address.

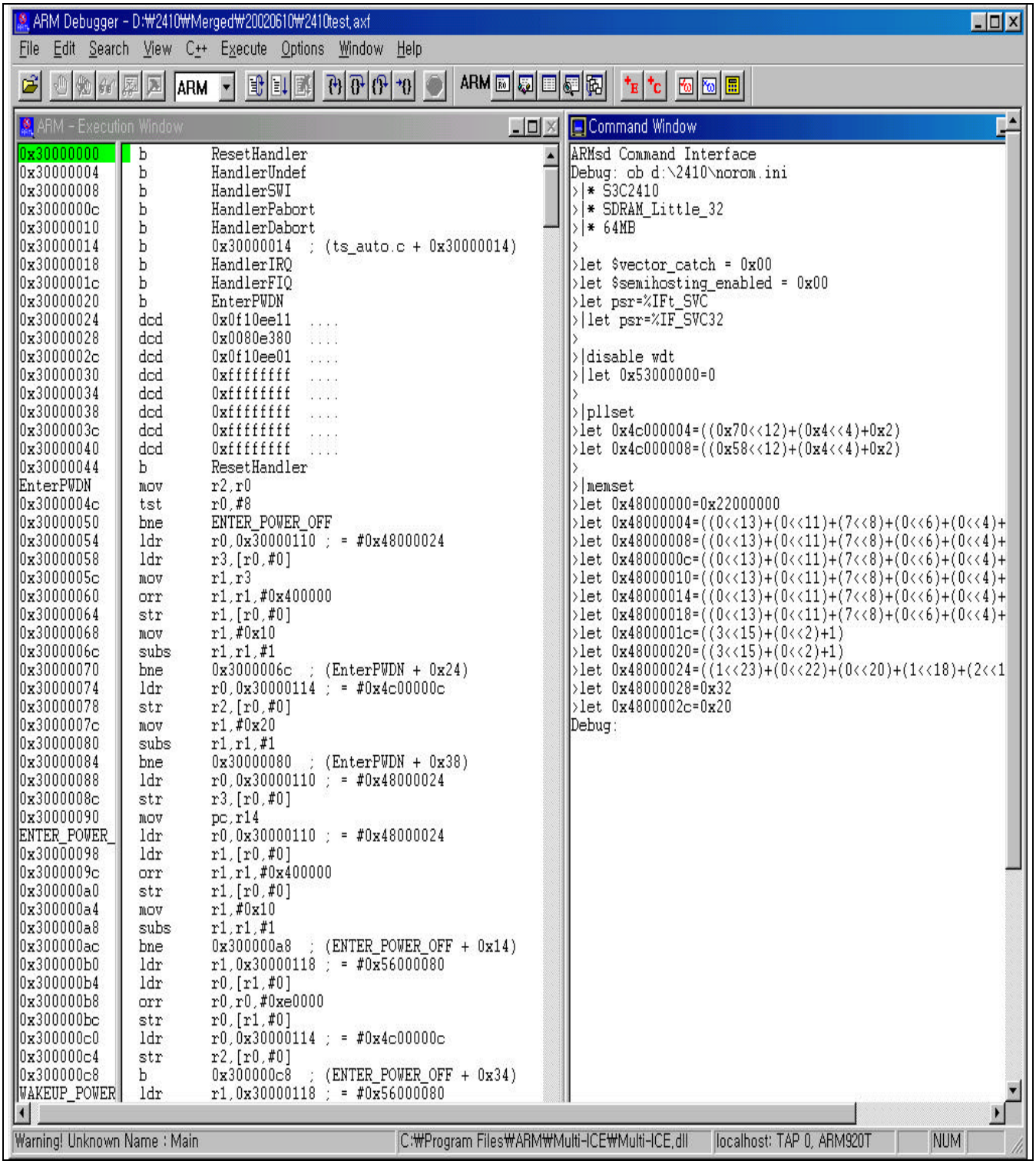
9. Turn the SMDK2410 off and again on.

WRITING IMAGE FILES TO INTEL STRATA FLASH MEMORY WITH MULTI-ICE

1. Connect MULTI-ICE and execute " norom.ini " file.

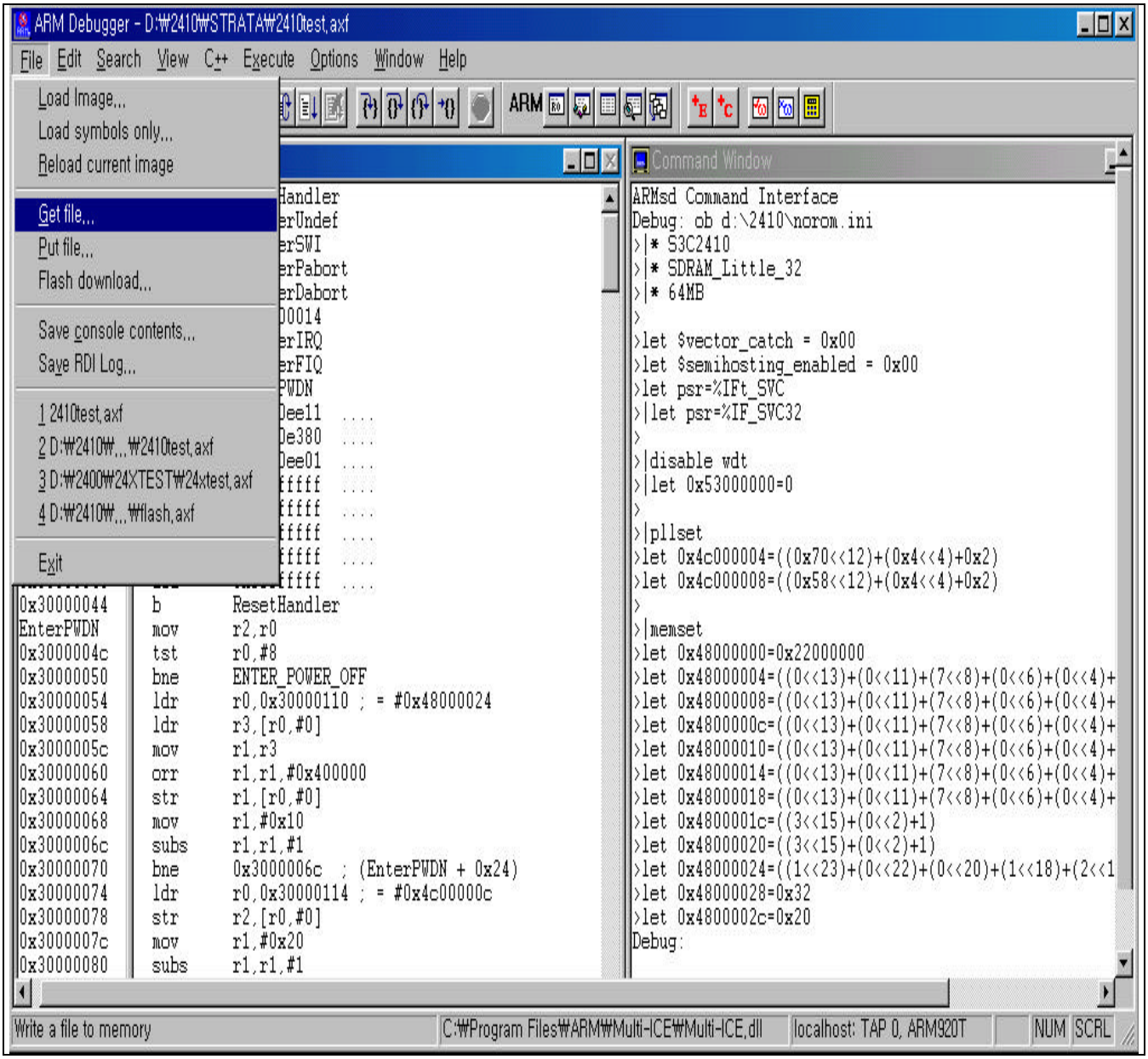


2. Load the image file (2410TEST.axf) to execute.

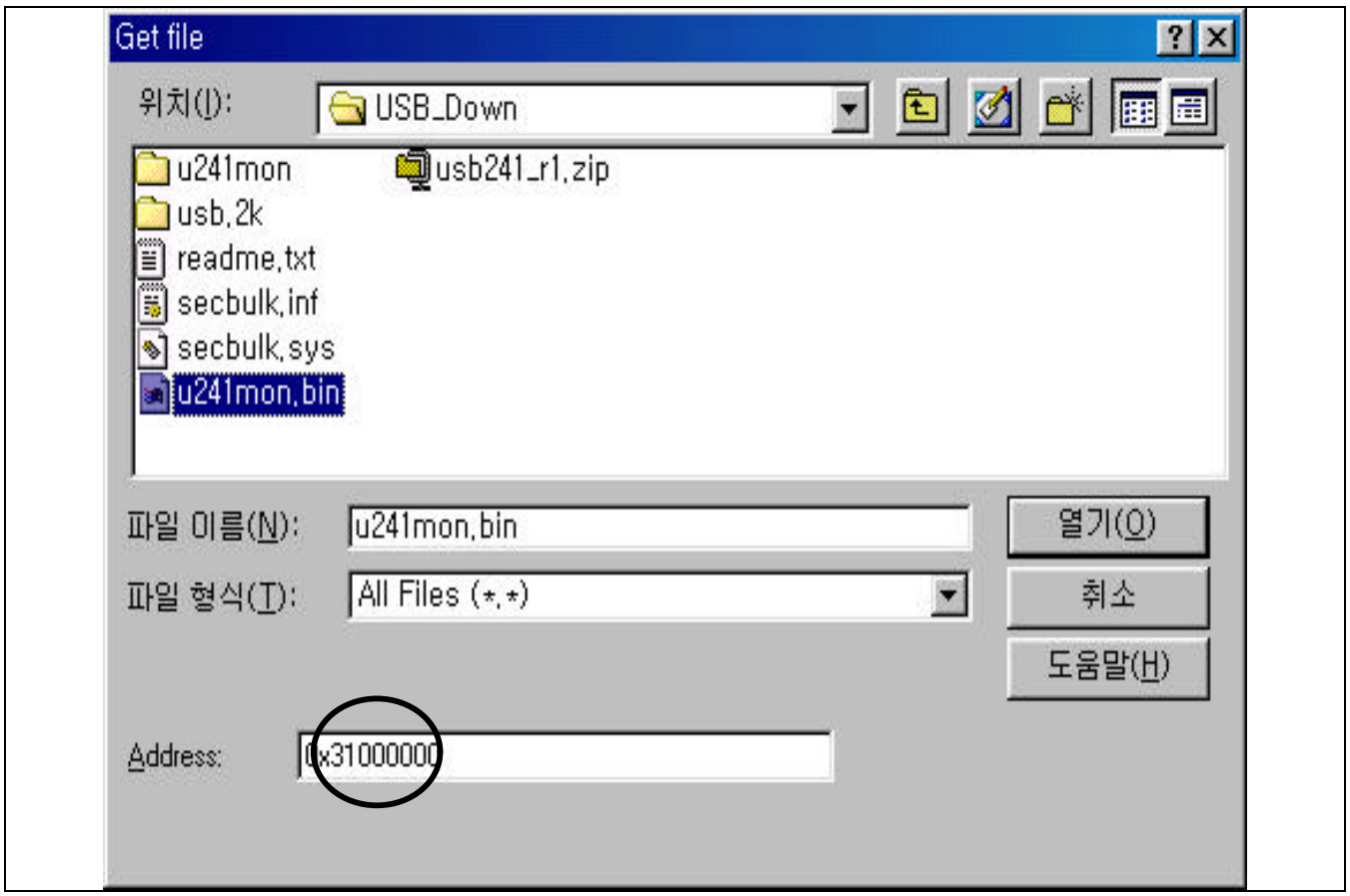


SAMSUNG ELECTRONICS

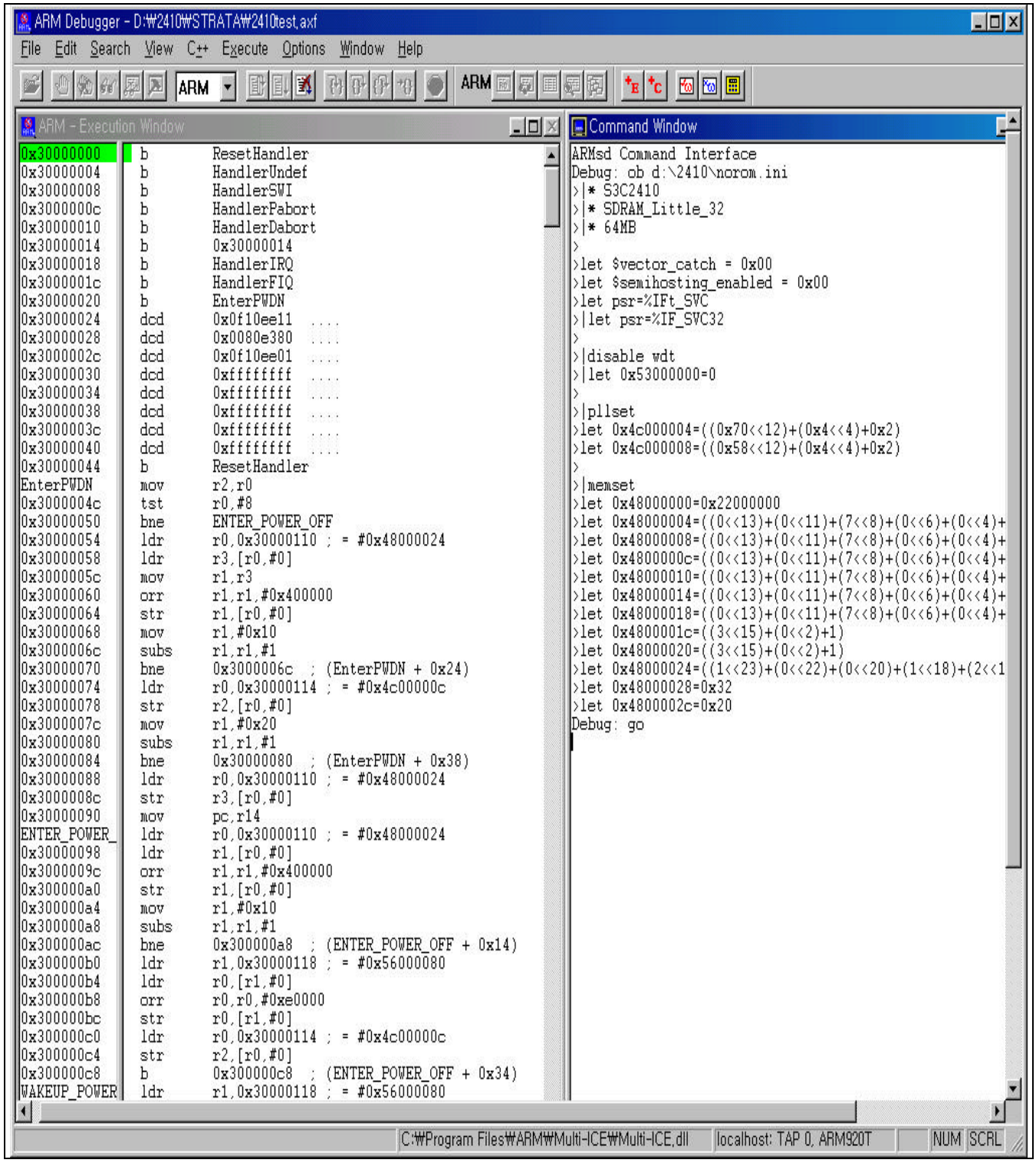
3. Get a target file written into INTEL STRATA flash memory.



— Get the target file to 0x31000000 in SMDK2410 Board.

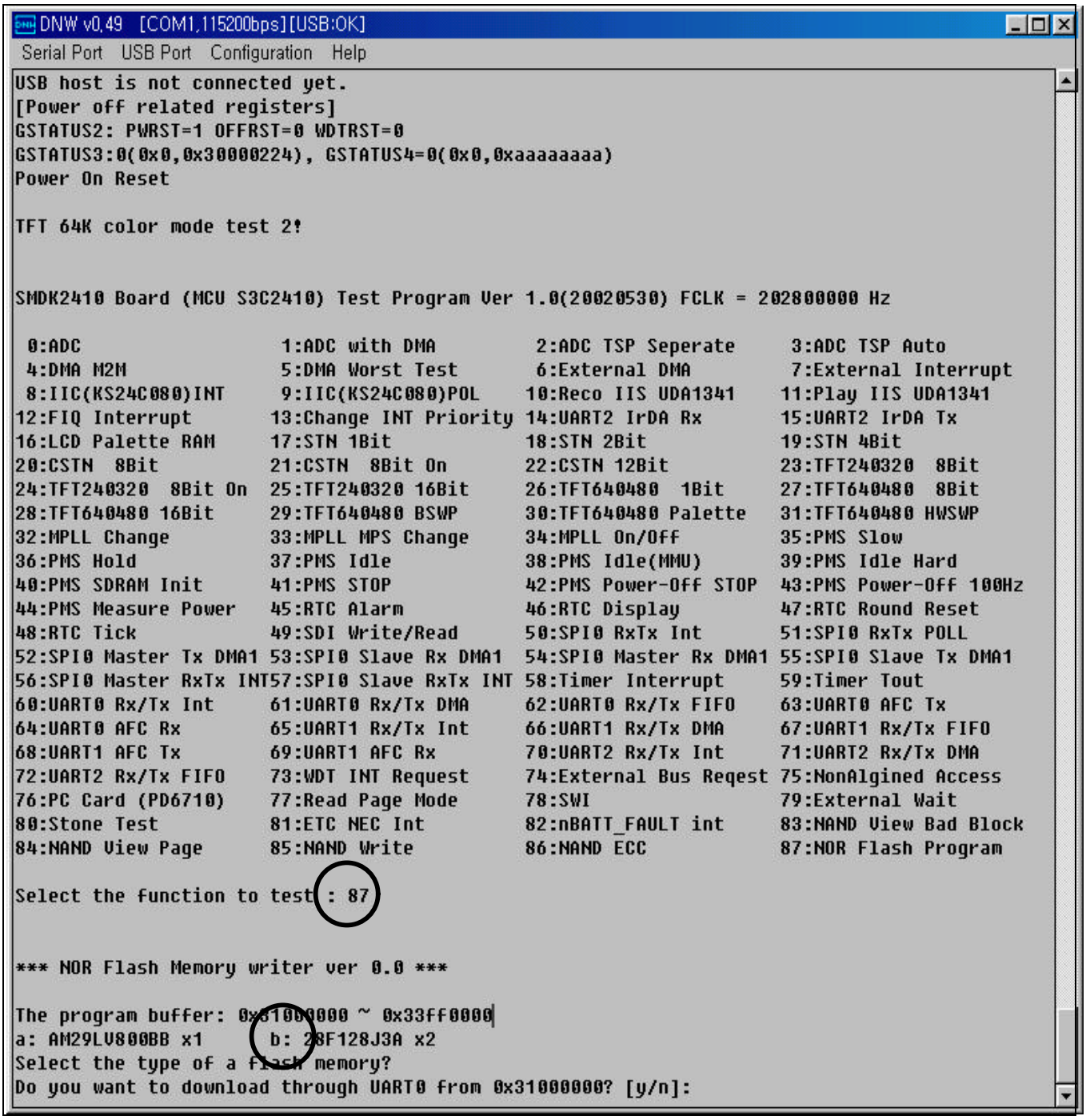


4. Execute 2410TEST.axf file with GO command.



5. Select " 87:NOR Flash Program " on the DNW.

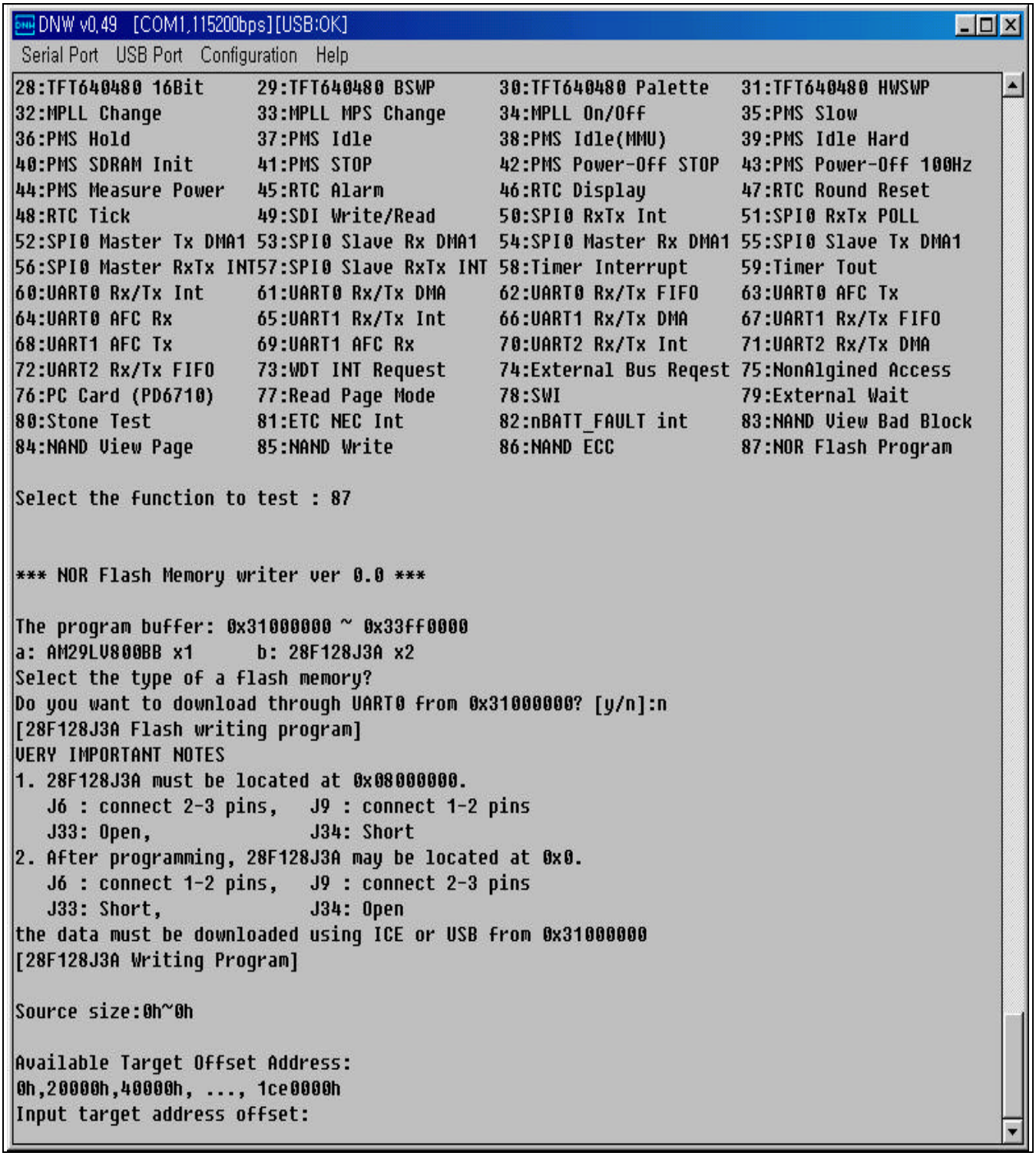
NOTE: If you want to download 2410TEST.bin without MULTI-ICE, then skip the 1, 2 & 3 steps above and download 2410TEST.bin using the DNW (See EXECUTE 2410TEST WITHOUT MULTI-ICE).
After downloading 2410TEST.bin with the DNW, then you can also see the figure below.



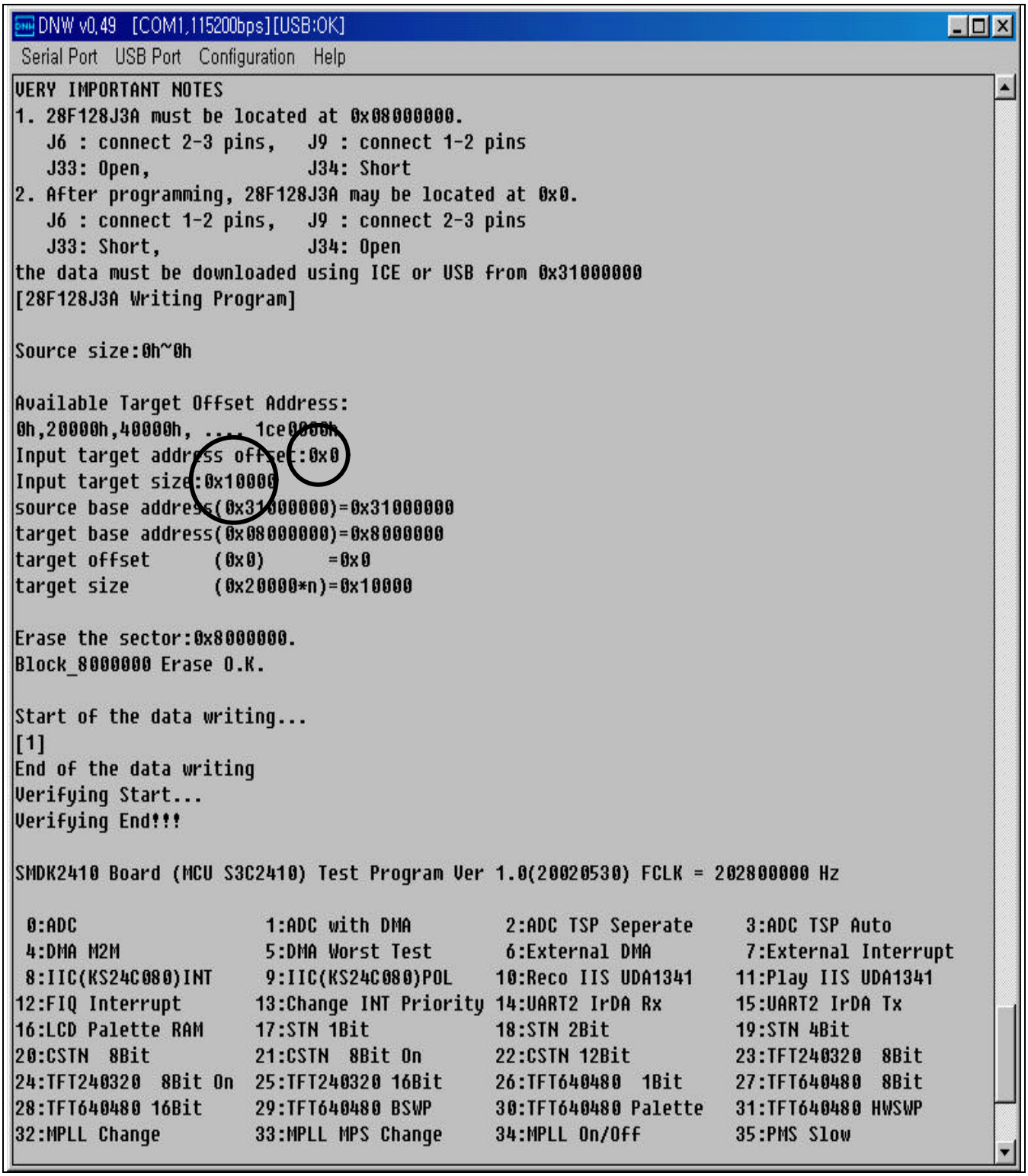
6. Select the type of memory as 28F128J3A (INTEL STRATA flash) by typing ‘b’.

7. Select whether you download through UART0 or MULTI-ICE.

— Type ‘n’ then you can see the figure below in the DNW.



8. Write input target address offset and size of the target file in hexadecimal.

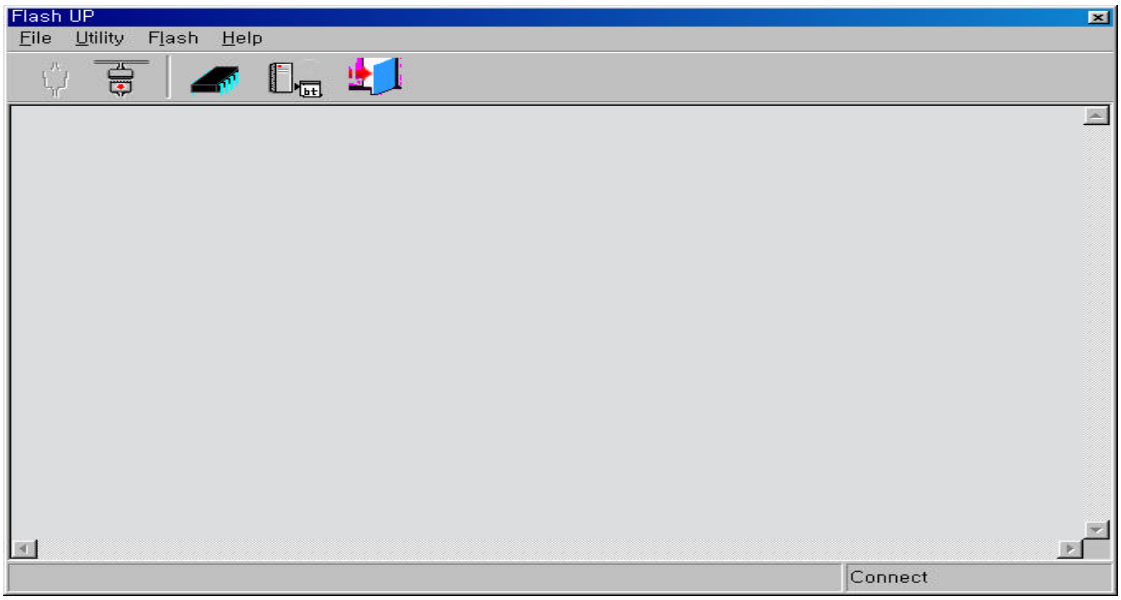


9. Turn the SMDK2410 off and again on.

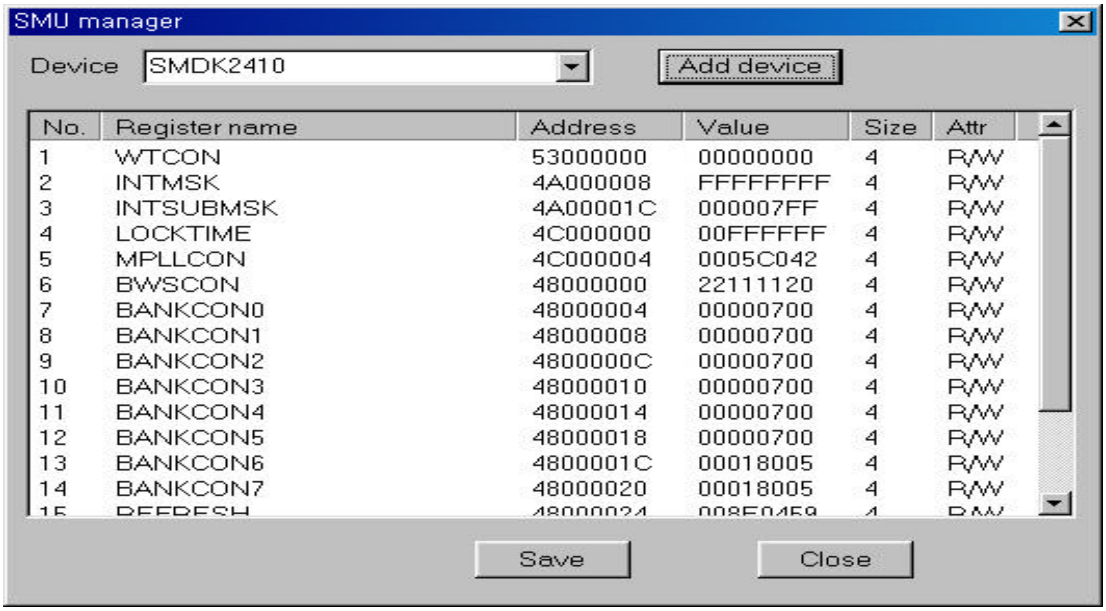
WRITING IMAGE FILES TO INTEL STRATA FLASH MEMORY WITH OPENICE32-A900

OPENice32-A900 can write image to Intel Strata Flash memory as Multi-ICE. However, OPENice32-A900 provide a Flash Write Program that is easy to use and don't require ARM SDT/ADS debugger nor DNW. For more information on the program, refer to OPENice32-A900 manual or contact AIJI System (www.aijisystem.com).

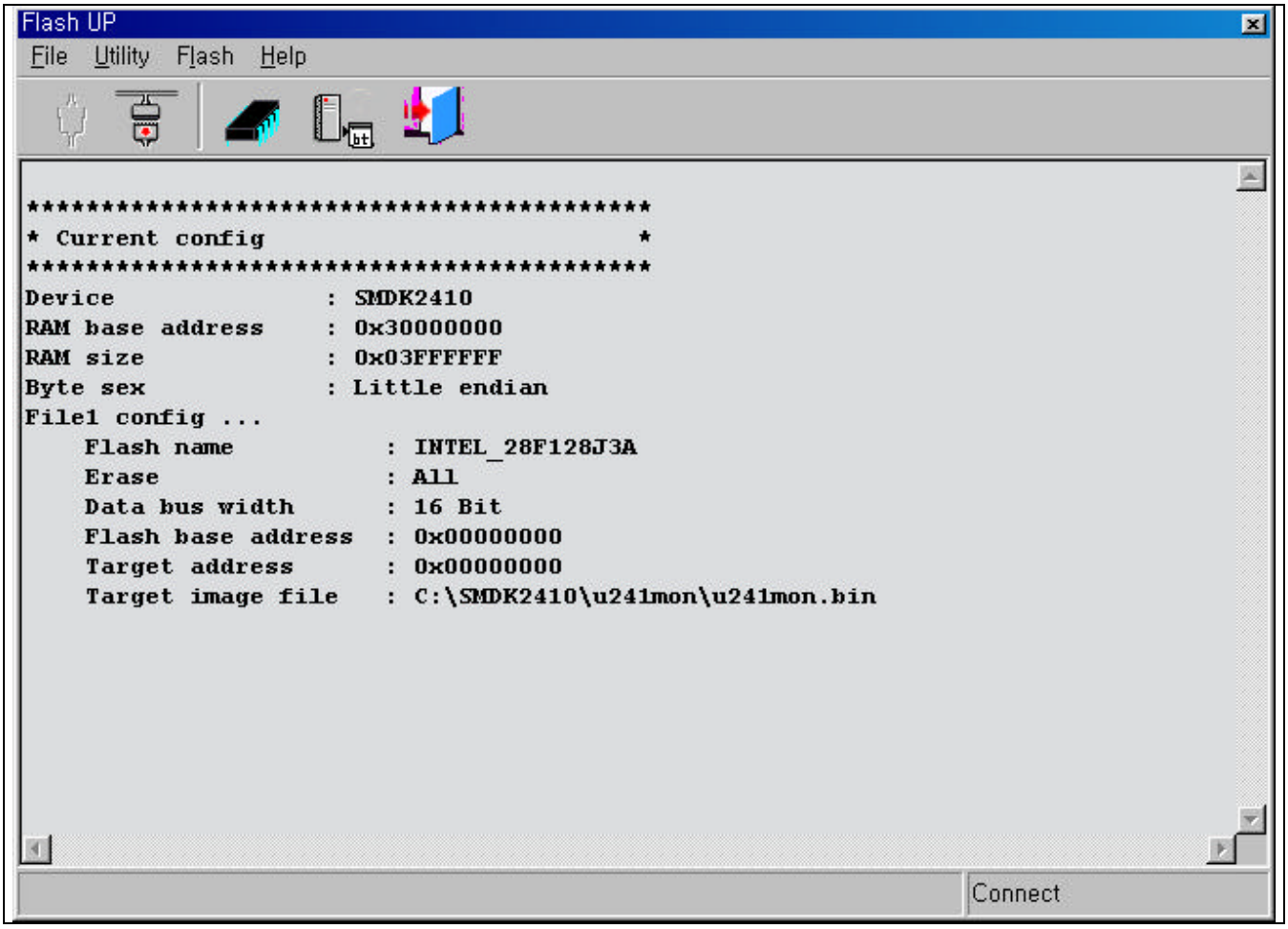
1. Connect OPENice32-A900 to PC through USB and to SMDK2410 board with 20pin Cable.
2. Set the Jumper J6, J9 as followings and switch on the board
   J6: 1-2 (short) J9: 2-3 (short)
3. Run the Flash Write program and select Connect MDS from the File menu.



4. Select SMU Manger from the utility menu and choose a device file, SMDK2410. It is used to initialize the system registers in case of there is no boot ROM. If you can't find the file, download the device file from AIJI System website (www.aijisystem.com). After that, edit each value if necessary.


SMU manager
Device
SMDK2410
Add device
No.
Register name
Address
Value
Size
Attr
1 WTCON 53000000 00000000 4 R/W
2 INTMSK 4A000008 FFFFFFFF 4 R/W
3 INTSUBMSK 4A00001C 000007FF 4 R/W
4 LOCKTIME 4C000000 00FFFFFF 4 R/W
5 MPLLCON 4C000004 0005C042 4 R/W
6 BWSCON 48000000 22111120 4 R/W
7 BANKCON0 48000004 00000700 4 R/W
8 BANKCON1 48000008 00000700 4 R/W
9 BANKCON2 4800000C 00000700 4 R/W
10 BANKCON3 48000010 00000700 4 R/W
11 BANKCON4 48000014 00000700 4 R/W
12 BANKCON5 48000018 00000700 4 R/W
13 BANKCON6 4800001C 00018005 4 R/W
14 BANKCON7 48000020 00018005 4 R/W
Save
Close

5. Select Config.. from the Flash menu and Set the write options as followings



— Device: SMDK2410

— Set SMU: Checked

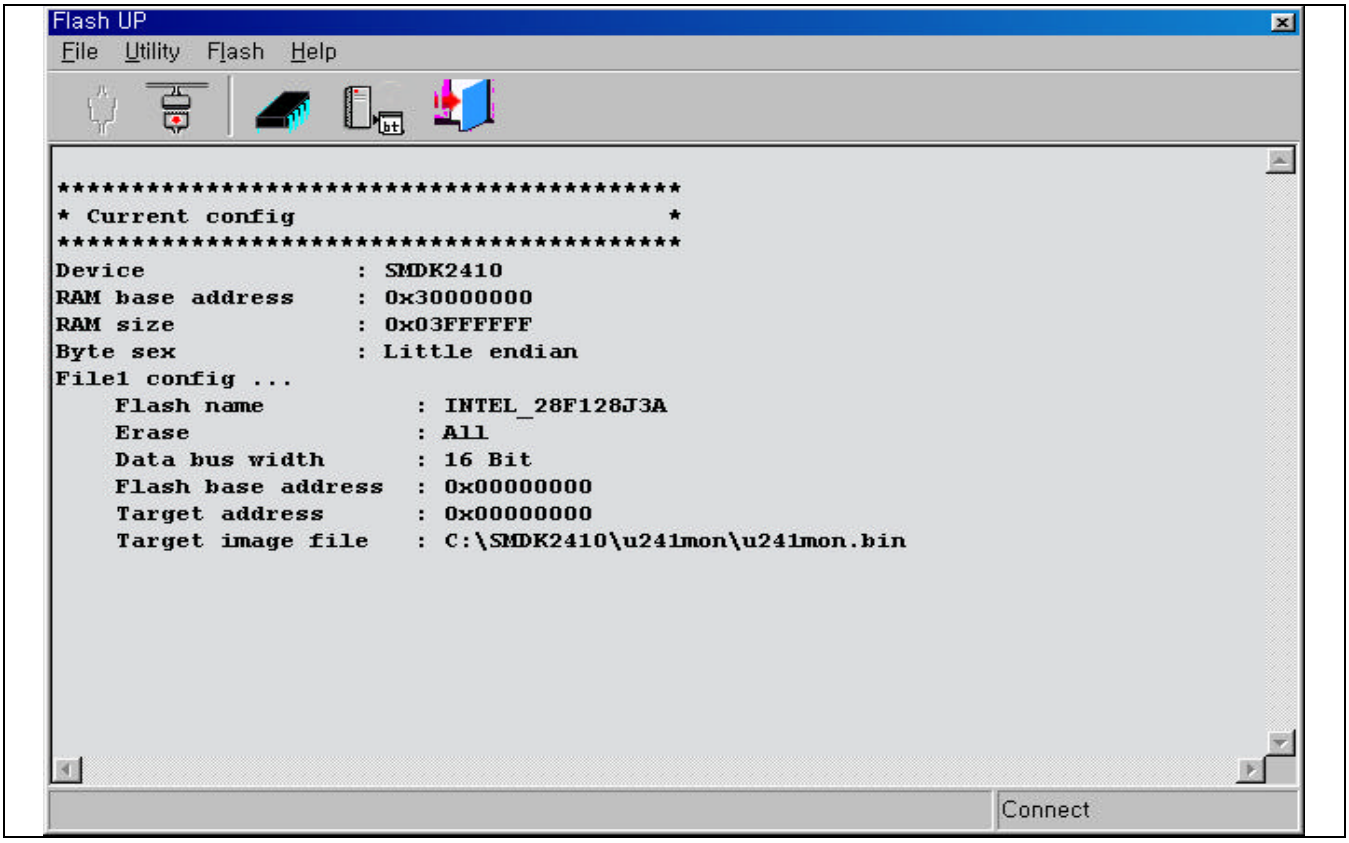
— RAM Information: Base Address:30000000 Size: 3FFFFFF

— Endian: Little

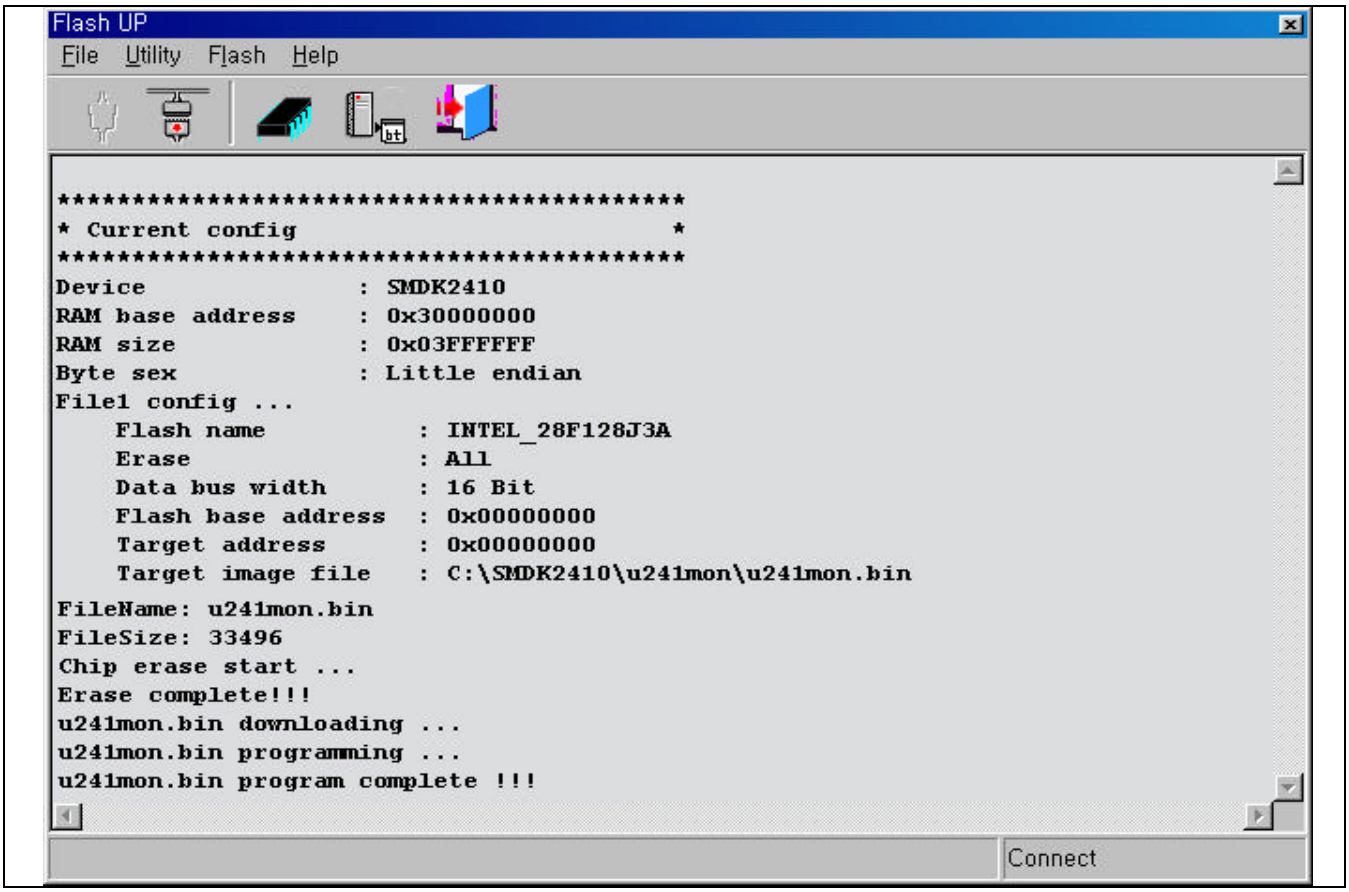
— File 1 page
Download: checked
Flash Device Name: INTEL_28F128J3A
Erase:Chip
Data Bus width: 16bit
Flash Address: Base Address: 0 Target Address:0

— Target Image File: u241mon.bin

6. Click OK. Then the current configuration is displayed in the window.



7. Select Write from the Flash Menu. Then it starts to erase the specified area of Intel Strata Flash and write the image to the Flash memory. It takes about 10 second.



4 SYSTEM DESIGN

OVERVIEW

The S3C2410X, SAMSUMG's 16/32-bit RISC microcontroller is cost-effective and high performance microcontroller solution for hand-held devices and general applications. The S3C2410X has the following integrated on-chip functions:

- 1.8V int., 3.3V memory, 3.3V external I/O microprocessor with 16KB I-Cache/16KB D-Cache/MMU
- External memory controller (SDRAM control and chip select logic)
- LCD controller (up to 4K color STN and 256K color TFT) with 1-ch LCD-dedicated DMA
- 4-ch DMAs with external request pins
- 3-ch UART (IrDA1.0, 16-Byte Tx FIFO, and 16-Byte Rx FIFO) / 2-ch SPI
- 1-ch multi-master IIC-BUS/1-ch IIS-BUS controller
- SD Host interface version 1.0 & Multi-Media Card Protocol version 2.11 compatible
- 2-port USB host /1- port USB device (ver 1.1)
- 4-ch PWM timers & 1-ch internal timer
- Watch Dog Timer
- 117-bit general purpose I/O ports / 24-ch external interrupt source
- Power control: Normal, Slow, Idle and Power-off mode
- 8-ch 10-bit ADC and touch screen interface
- RTC with calendar function
- On-chip clock generator with PLL

APPLICABLE SYSTEM WITH S3C2410X

The S3C2410X, SAMSUMG's 16/32-bit RISC microcontroller offers various functions and high efficiencies. In addition to the high performance, the S3C2410X offers low current consumption, ensuring low costs.
The followings are sample applications that can be designed with the S3C2410X:

- GPS
- Personal Data Assistance (PDA)
- Fish Finder
- Portable Game Machine
- Fingerprint Identification System
- Car Navigation System
- Smart Phone
- Mobile Information Terminal (MIT)
- Web Screen Phone
- Web Pad

MEMORY INTERFACE DESIGN

BOOT ROM DESIGN

After the system reset, the S3C2410X accesses 0x00000000 address, configuring some system variables. Therefore, this special code (boot ROM image) should be located on the address 0x00000000. Bus width of boot ROM can be selected by setting OM[1:0] pins.

Table 4-1. Data Bus Width for ROM Bank 0

OM[1:0]	Data Bus Width
00	NAND boot
01	16-bit (halfword)
10	32-bit (word)
11	Test mode

NAND BOOT DESIGN

Figure 4-1 shows a design with NAND boot.

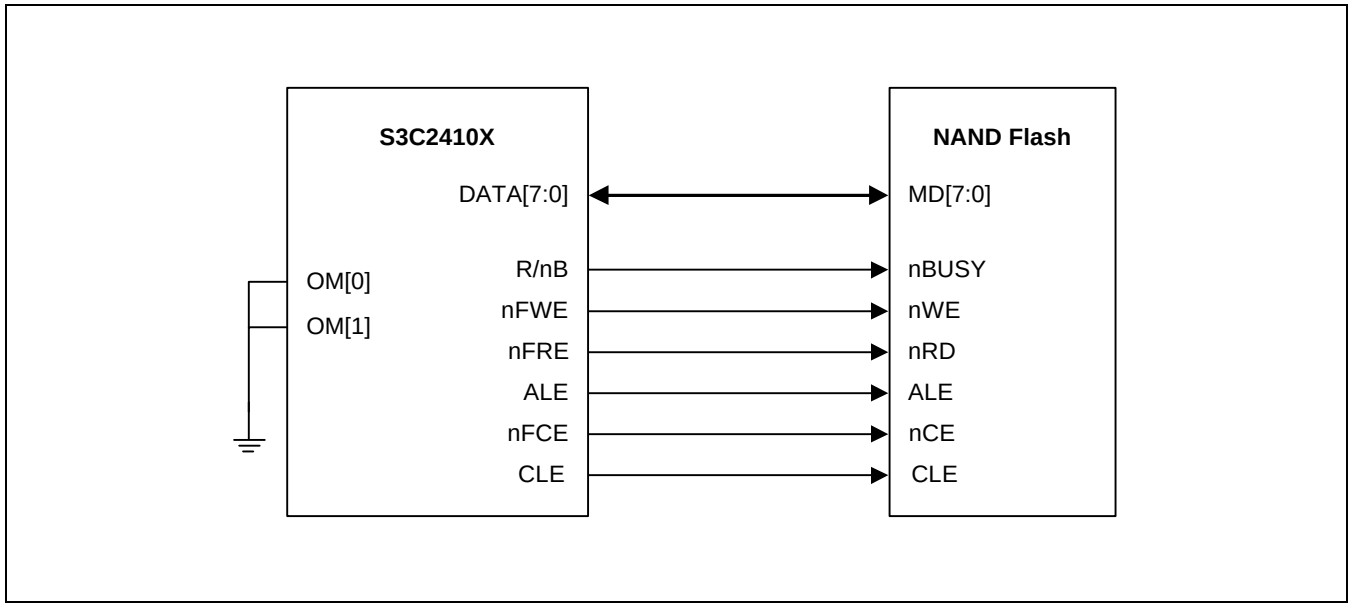


Figure 4-1. NAND Boot Design

MAKING NAND BOOT IMAGE

When making a NAND boot loader image, you can use the binary file that is made from compiling and linking.

HALFWORD BOOT ROM DESIGN WITH BYTE EEPROM/FLASH

Figure 4-2 shows a design with halfword boot ROM with byte EEPROM/Flash.

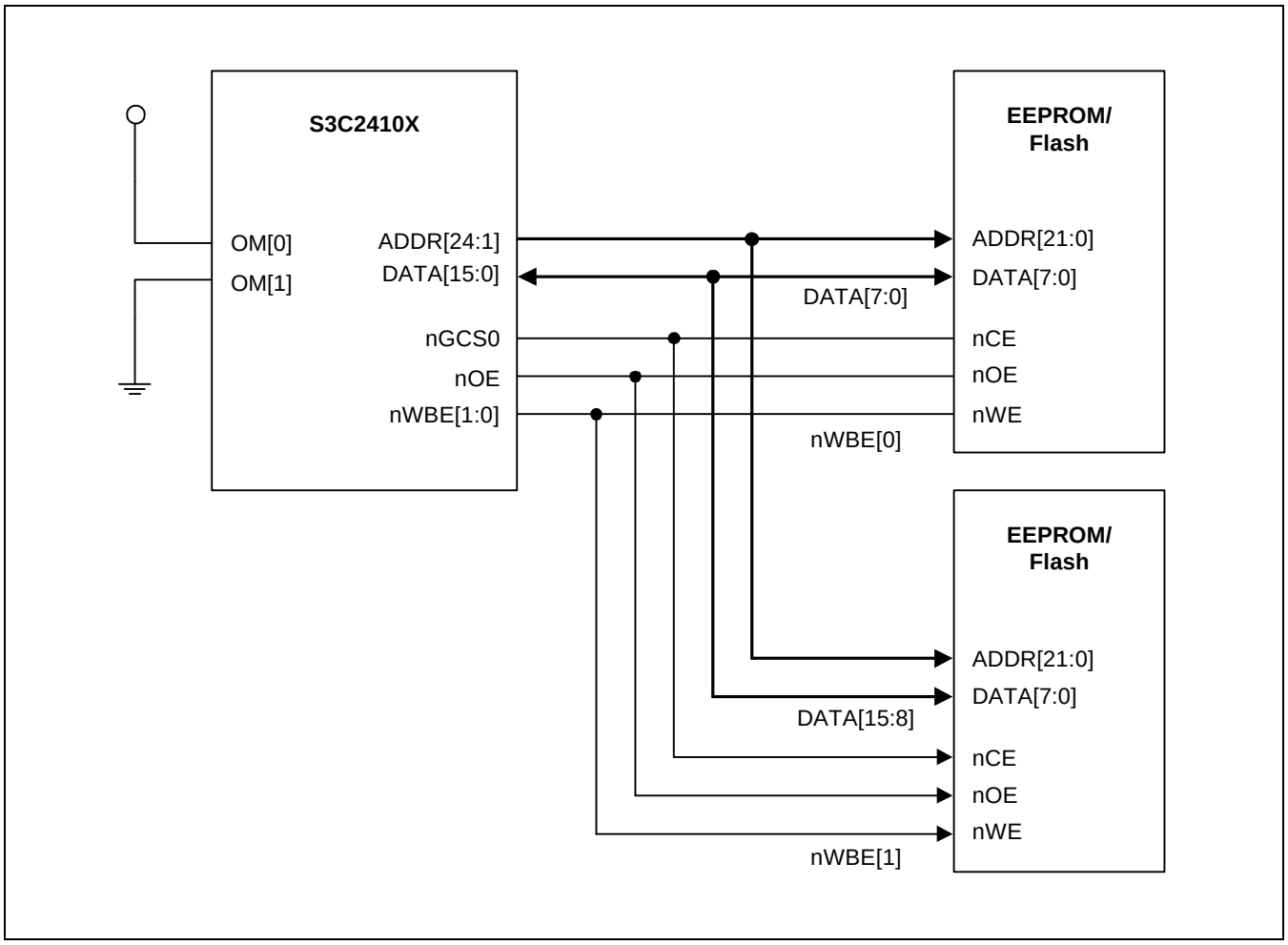


Figure 4-2. Halfword Boot ROM Design with Byte EEPROM/Flash

MAKING HALFWORD ROM IMAGE WITH BYTE EEPROM/FLASH

When make halfword ROM image, you can split two image files, EVEN and ODD.

Table 4-2. Relationship ROM Image and Endian

	Big Endian	Little Endian
DATA[7:0]	Odd	Even
DATA[15:8]	Even	Odd

HALFWORD BOOT ROM DESIGN WITH HALFWORD EEPROM/FLASH

Figure 4-3 shows a design with halfword boot ROM with byte EEPROM/Flash.

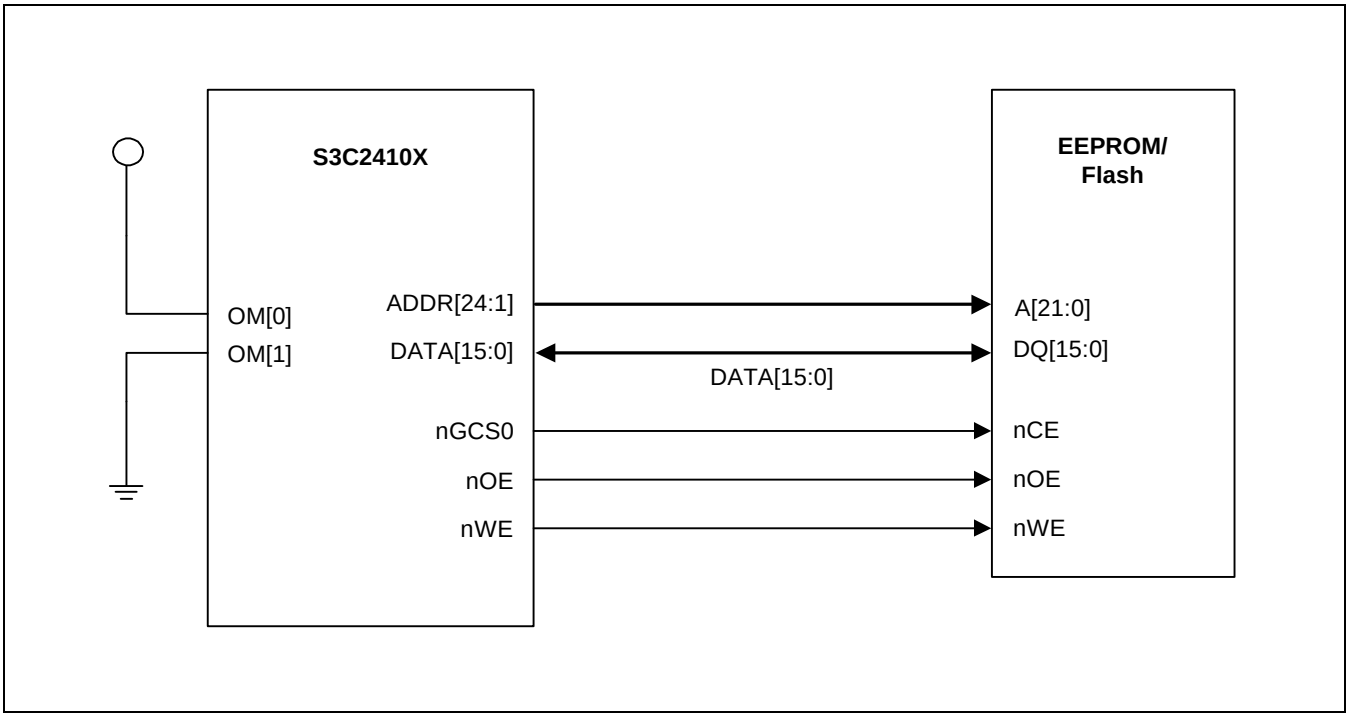


Figure 4-3. The Halfword Boot ROM Design with Halfword EEPROM/Flash

WORD BOOT ROM DESIGN WITH BYTE EEPROM/FLASH

Figure 4-4 shows a design with word boot ROM with byte EEPROM/Flash.

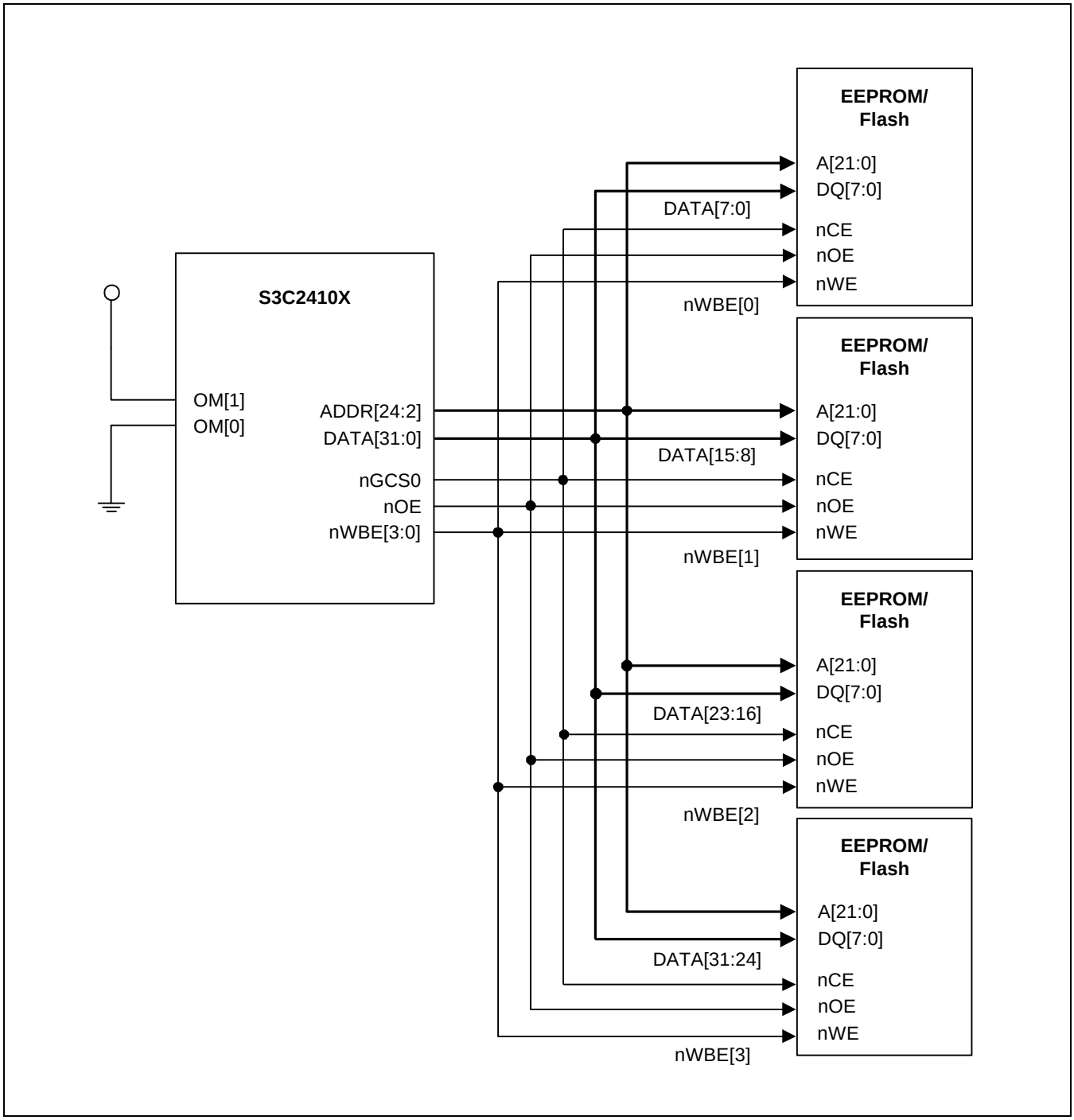
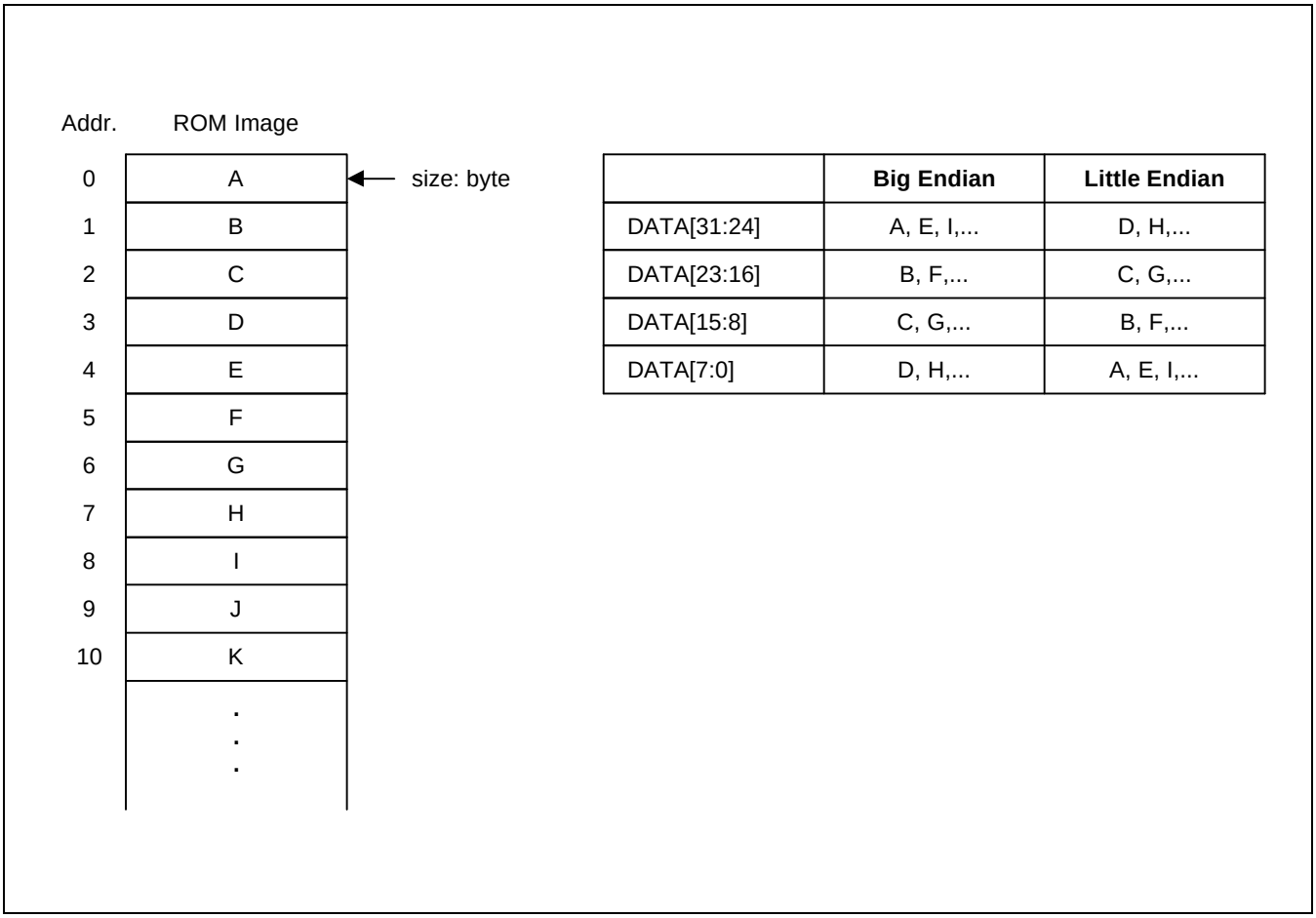


Figure 4-4. The Word Boot ROM Design with Byte EEPROM/Flash

MAKING WORD ROM IMAGE WITH BYTE EEPROM/FLASH

When you make a word ROM image, you can split it into four image files.



	Big Endian	Little Endian
DATA[31:24]	A, E, I,...	D, H,...
DATA[23:16]	B, F,...	C, G,...
DATA[15:8]	C, G,...	B, F,...
DATA[7:0]	D, H,...	A, E, I,...

Figure 4-5. Relationship of ROM Image and Endian

MEMORY BANK DESIGN AND CONTROL

The S3C2410X has six ROM/SRAM banks (including BANK0 for boot ROM) and two ROM/SRAM/SDRAM banks. The system manager on the S3C2410X can control access time, data bus width for each bank by S/W. The access time of ROM/SRAM banks and SDRAM banks is controlled by BANKCON0~5 and BANKCON6~7 control register on the system manager. The data bus width for each ROM/SRAM banks is controlled by BWSCON control register.

The ROM bank 0 is used for boot ROM bank, therefore data bus width of bank 0 is controlled by H/W. OM[1:0] is used for this purpose.

The control of BWSCON, BANKCON0-7, REFRESH, BANKSIZE, and MRSRB6/7 is performed during the system reset. A sample code for special register configuration is described below.

Sample code for special register configuration

```
        ;Set memory control registers
        LDR     r0,=SMRDATA
        LDR     r1,=BWSCON      ;BWSCON Address
        ADD     r2, r0, #52     ;End address of SMRDATA
0
        LDR     r3, [r0], #4
        STR     r3, [r1], #4
        CMP     r2, r0
        BNE     %B0
         .
         .
         .
         .
         .
         .
SMRDATA
        DCD 0x22111120          ;BWSCON
        DCD 0x00000700          ;GCS0
        DCD 0x00000700          ;GCS1
        DCD 0x00000700          ;GCS2
        DCD 0x00000700          ;GCS3
        DCD 0x00000700          ;GCS4
        DCD 0x00000700          ;GCS5
        DCD 0x00018005          ;GCS6 SDRAM(Trcd=3,SCAN=9)
        DCD 0x00018005          ;GCS7 SDRAM(Trcd=3,SCAN=9)
        DCD 0x008e0000+1113;Refresh(REFEN=1,TREFMD=0,Trp=2 clk,
                                ;       Trc=7 clk, Tchr=3 clk,Ref CNT)
        DCD 0x32                ;Bank size, 128MB/128MB
        DCD 0x30                ;MRSR 6(CL=3 clk)
        DCD 0x30                ;MRSR 7(CL=3 clk)
```

ROM/SRAM BANK DESIGN

The ROM/SRAM banks 1-7 can have a variety of width of data bus, and the bus width is controlled by S/W. A sample design for ROM/SRAM bank 1-7 is shown in Figure 4-6, Figure 4-7, Figure 4-8 and Figure 4-9.

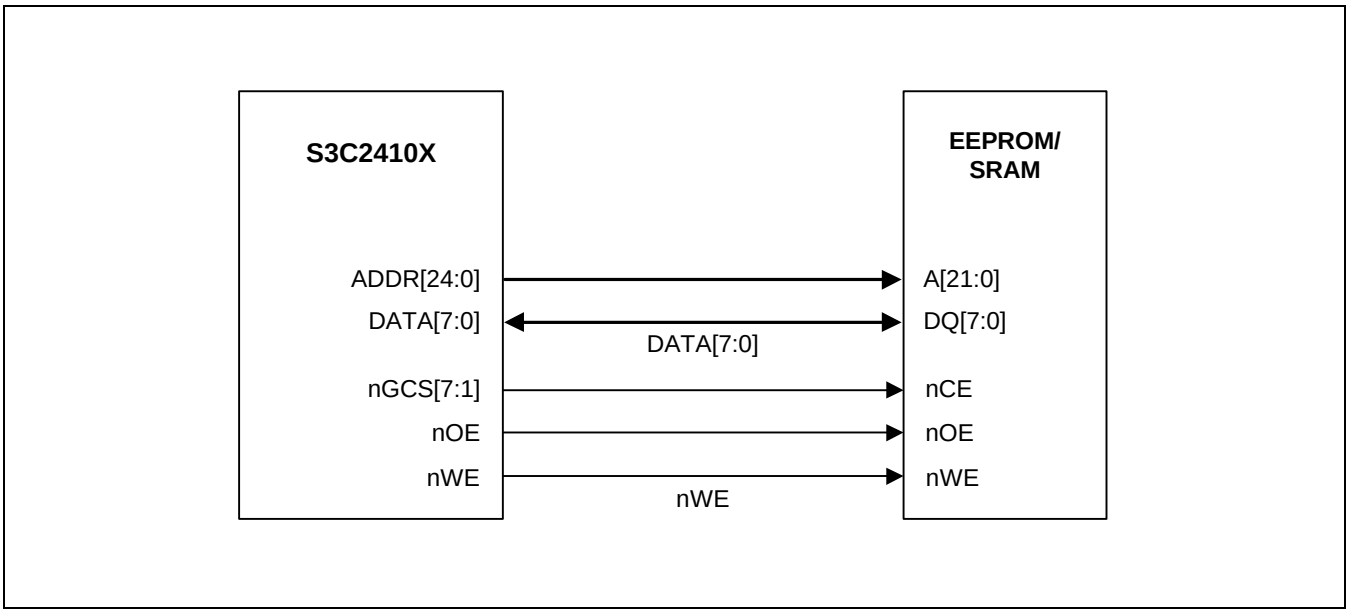


Figure 4-6. One-byte EEPROM/SRAM Bank Design

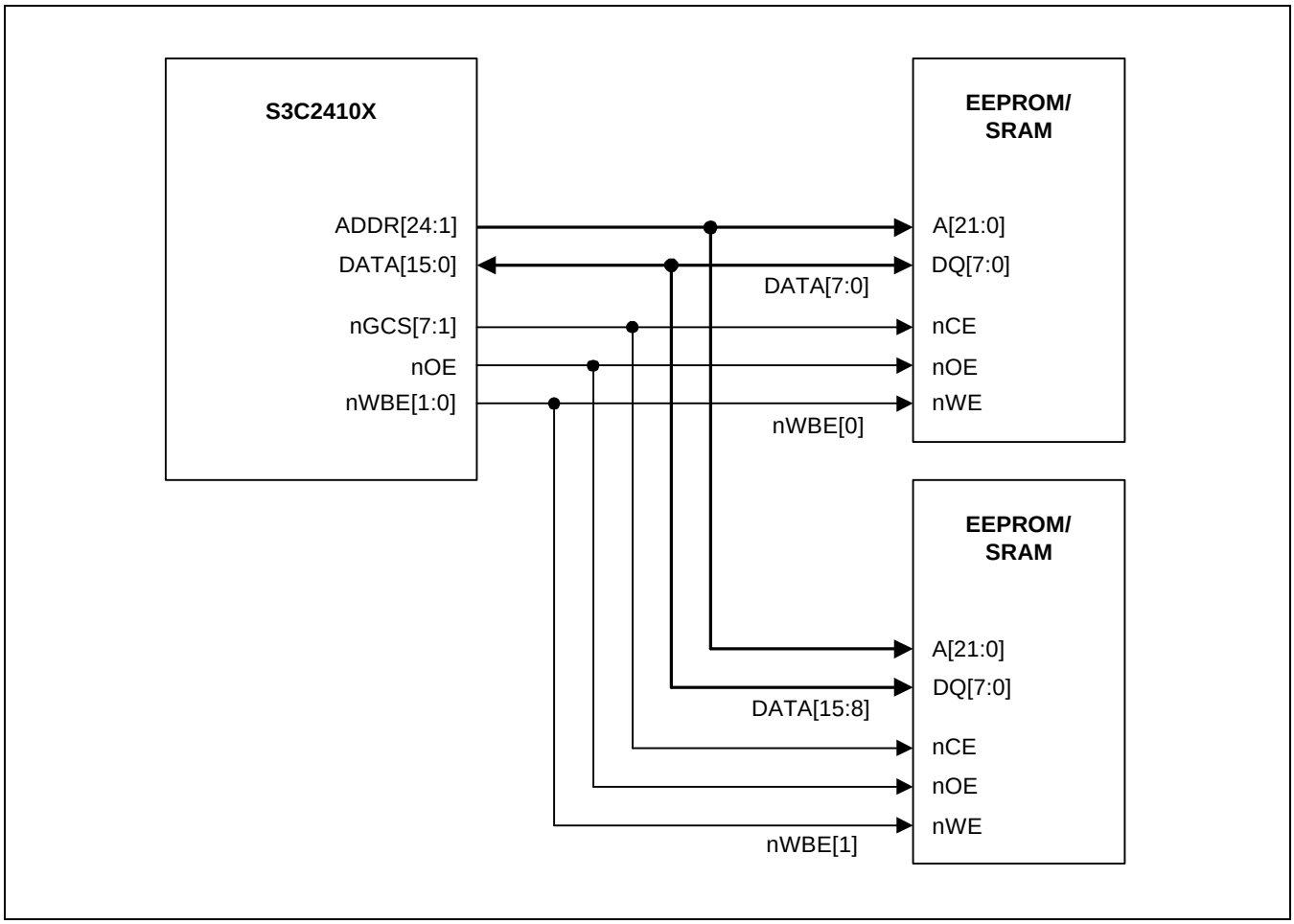


Figure 4-7. HalfWword EEPROM/SRAM Bank Design

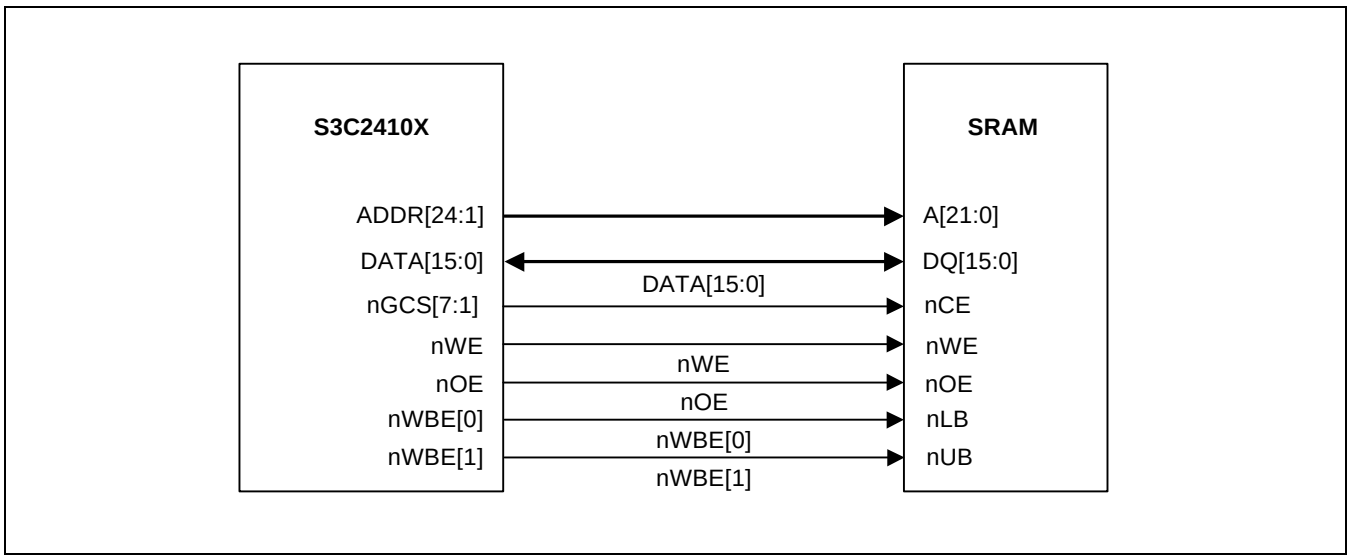


Figure 4-8. Halfword SRAM Bank Design with Halfword SRAM

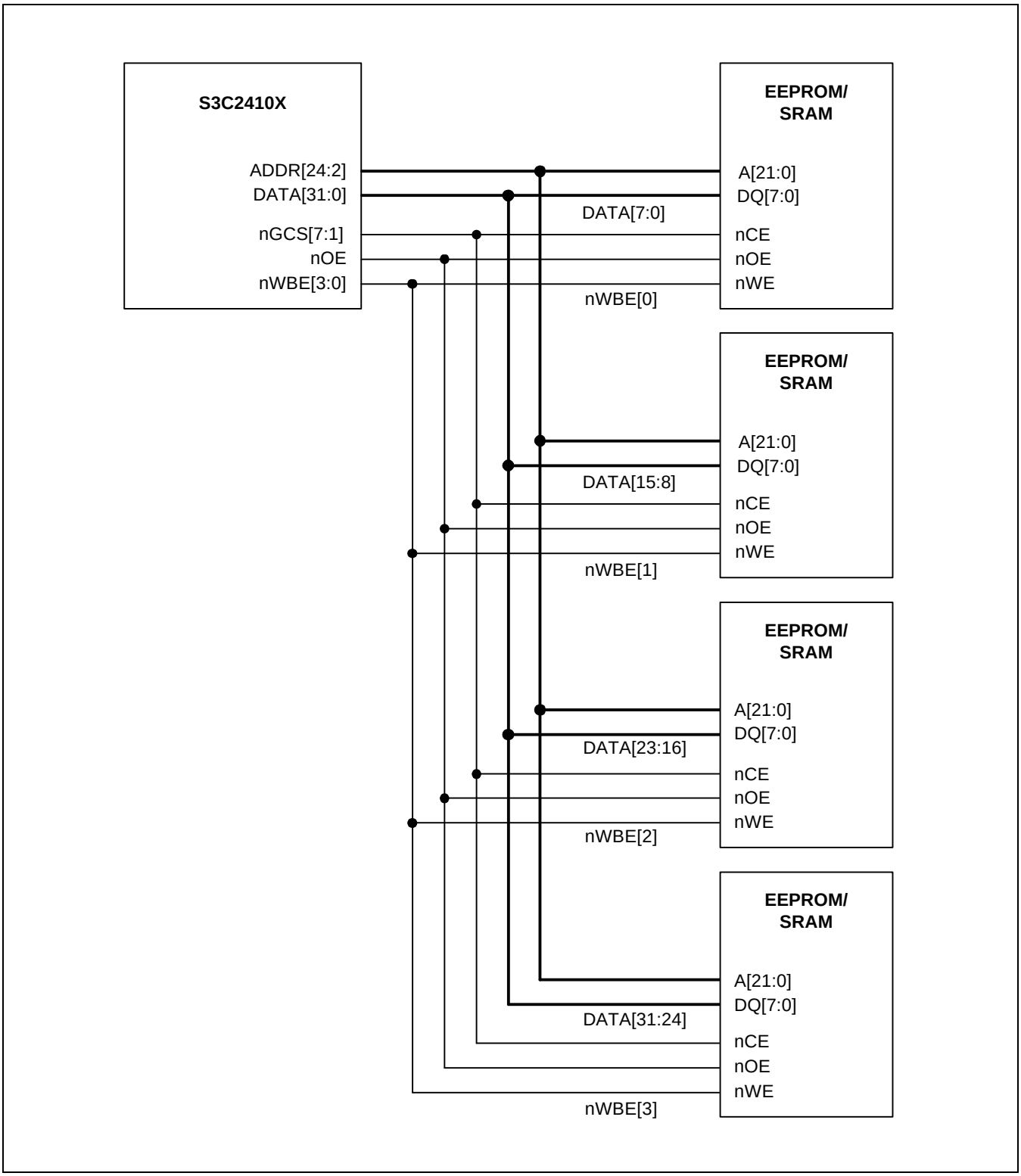


Figure 4-9. Word EEPROM/SRAM Bank Design

SDRAM BANK DESIGN FOR S3C2410X

Table 4-3. SDRAM Bank Address configuration

Bank Size	Bus Width	Base Component	Memory Configuration	Bank Address
2MByte	x8	16Mbit	(1M x 8 x 2Bank) x 1	A20
	x16		(512K x 16 x 2B) x 1	
4MB	x8	16Mb	(2M x 4 x 2B) x 2	A21
	x16		(1M x 8 x 2B) x 2	
	x32		(512K x 16 x 2B) x 2	
8MB	x16	16Mb	(2M x 4 x 2B) x 4	A22
	x32		(1M x 8x 2B) x 4	
	x8	64Mb	(4M x 8 x 2B) x 1	
	x8		(2M x 8 x 4B) x 1	A[22:21]
	x16		(2M x 16 x 2B) x 1	A22
	x16		(1M x 16 x 4B) x 1	A[22:21]
	x32		(512K x 32 x 4B) x 1	
16MB	x32	16Mb	(2M x 4 x 2B) x 8	A23
	x8	64Mb	(8M x 4 x 2B) x 2	
	x8		(4M x 4 x 4B) x 2	A[23:22]
	x16		(4M x 8 x 2B) x 2	A23
	x16		(2M x 8 x 4B) x 2	A[23:22]
	x32		(2M x 16 x 2B) x 2	A23
	x32		(1M x 16 x 4B) x 2	A[23:22]
	x8	128Mb	(4M x 8 x 4B) x 1	
	x16		(2M x 16 x 4B) x 1	
32MB	x16	64Mb	(8M x 4 x 2B) x 4	A24
	x16		(4M x 4 x 4B) x 4	A[24:23]
	x32		(4M x 8 x 2B) x 4	A24
	x32		(2M x 8 x 4B) x 4	A[24:23]
	x16	128Mb	(4M x 8 x 4B) x 2	
	x32		(2M x 16 x 4B) x 2	
	x8	256Mb	(8M x 8 x 4B) x 1	
	x16		(4M x 16 x 4B) x 1	

Table 4-3. SDRAM Bank Address configuration (Continued)

Bank Size	Bus Width	Base Component	Memory Configuration	Bank Address
64MB	x32	128Mb	(4M x 8 x 4B) x 4	A[25:24]
	x16	256Mb	(8M x 8 x 4B) x 2	
	x32		(4M x 16 x 4B) x 2	
	x8	512Mb	(16M x 8 x 4B) x 1	
128MB	x32	256Mbit	(8M x 8 x 4Bank) x 4	A[26:25]
	x8	512Mb	(32M x 4 x 4B) x 2	
	x16		(16M x 8 x 4B) x 2	

The required SDRAM interface pin is CKE, SCLK, nSCS[1:0], nSCAS, nSRAS, DQM[3:0] and ADDR[12]/AP. The sample design with SDRAM is shown in Figure 4-10 and Figure 4-11.

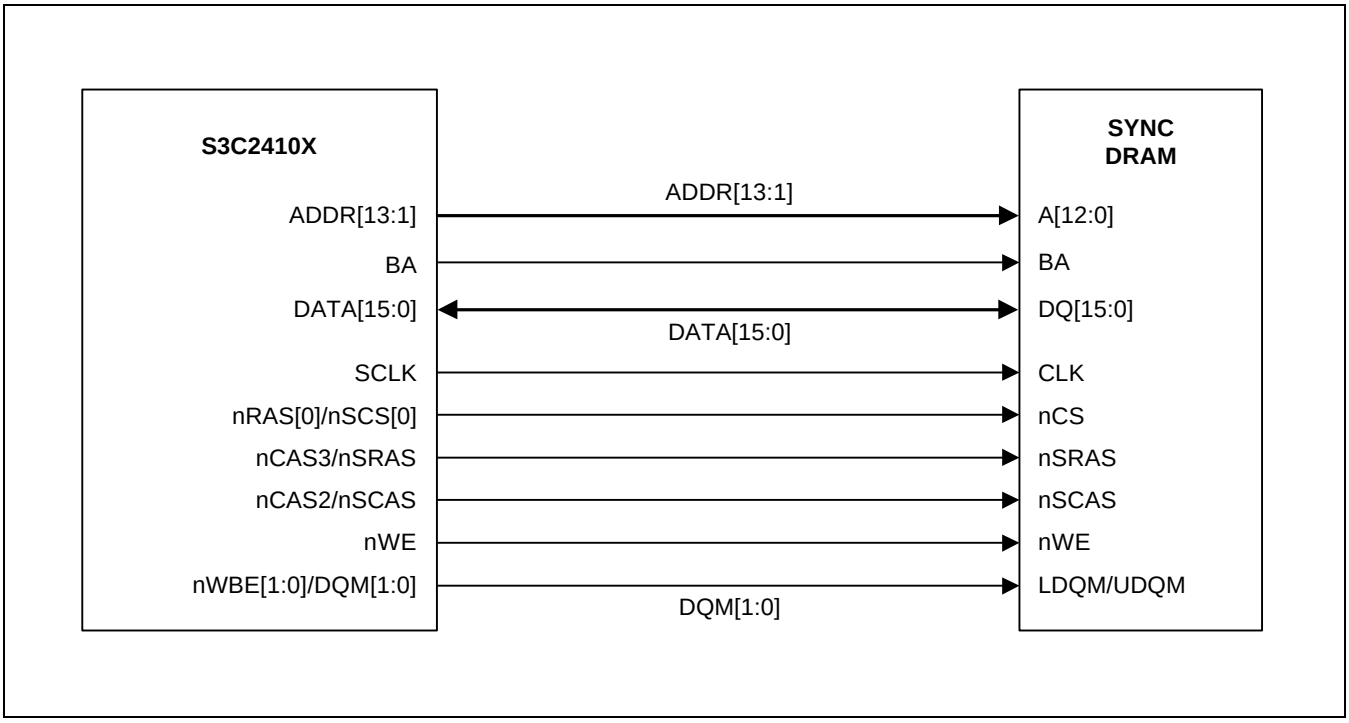


Figure 4-10. Halfword SDRAM Design with Halfword Component

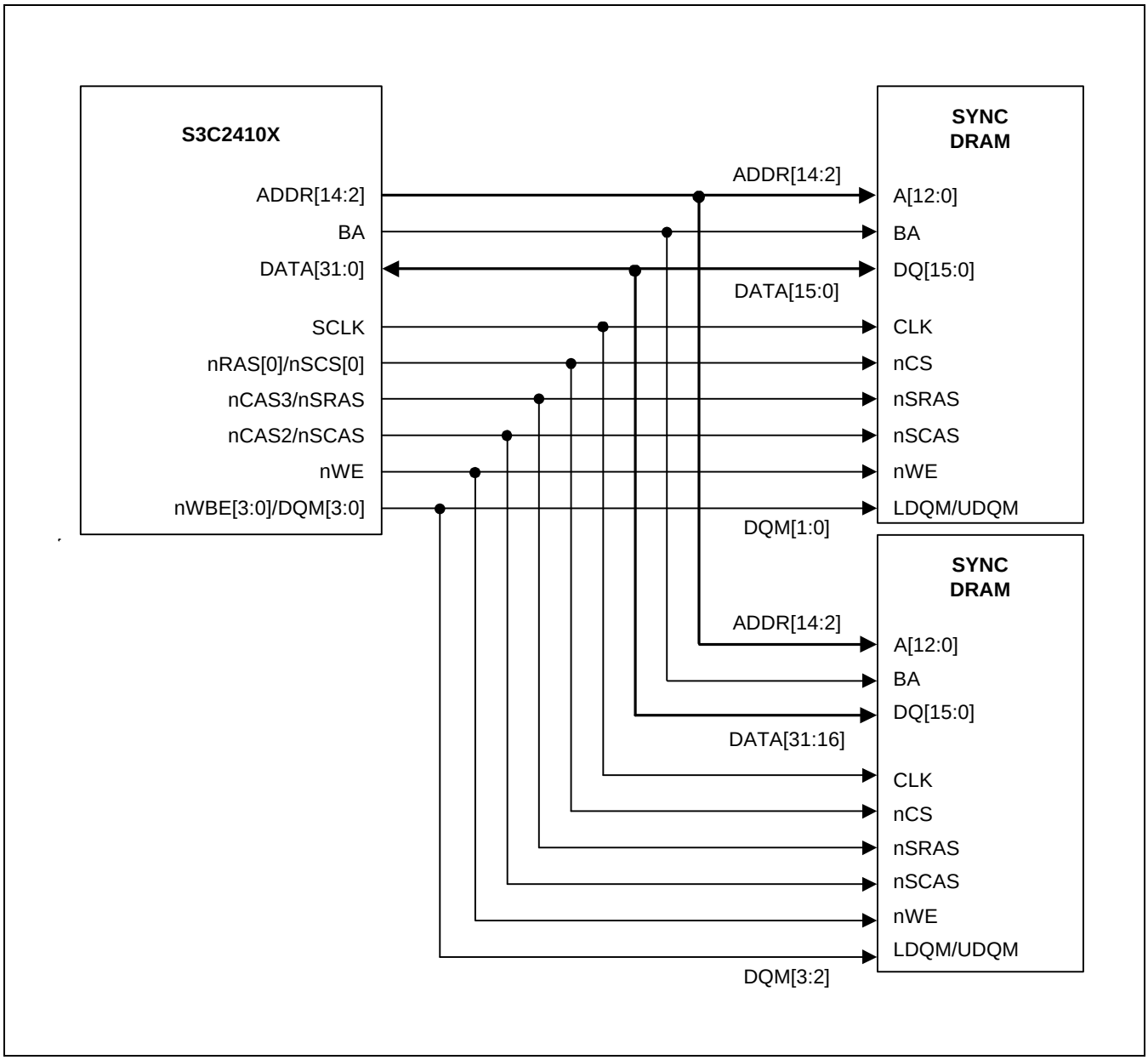


Figure 4-11. Word SDRAM Design with Halfword Component

GPC CARD (GPCMCIA) INTERFACE APPLICATION USING CL-PD6710 (CIRRUS LOGIC)

The GPC card (GPCMCIA card) can be interfaced with S3C2410X using following components:

- CL-PD6710 from Cirrus logic
- TPS2211 from Texas Instruments
- Refer to end of this chapter, which has sample codes of GPCMCIA application.

We tested the GPC card interface by accessing the card information structure (CIS) in the modem card as Figure 4-12. The following test code and schematics are attached in the end of this chapter.

File Name	File Descriptions
pd6710.h	CL-PD6710 register definitions
pd6710.c	CL-PD6710 GPC Card program

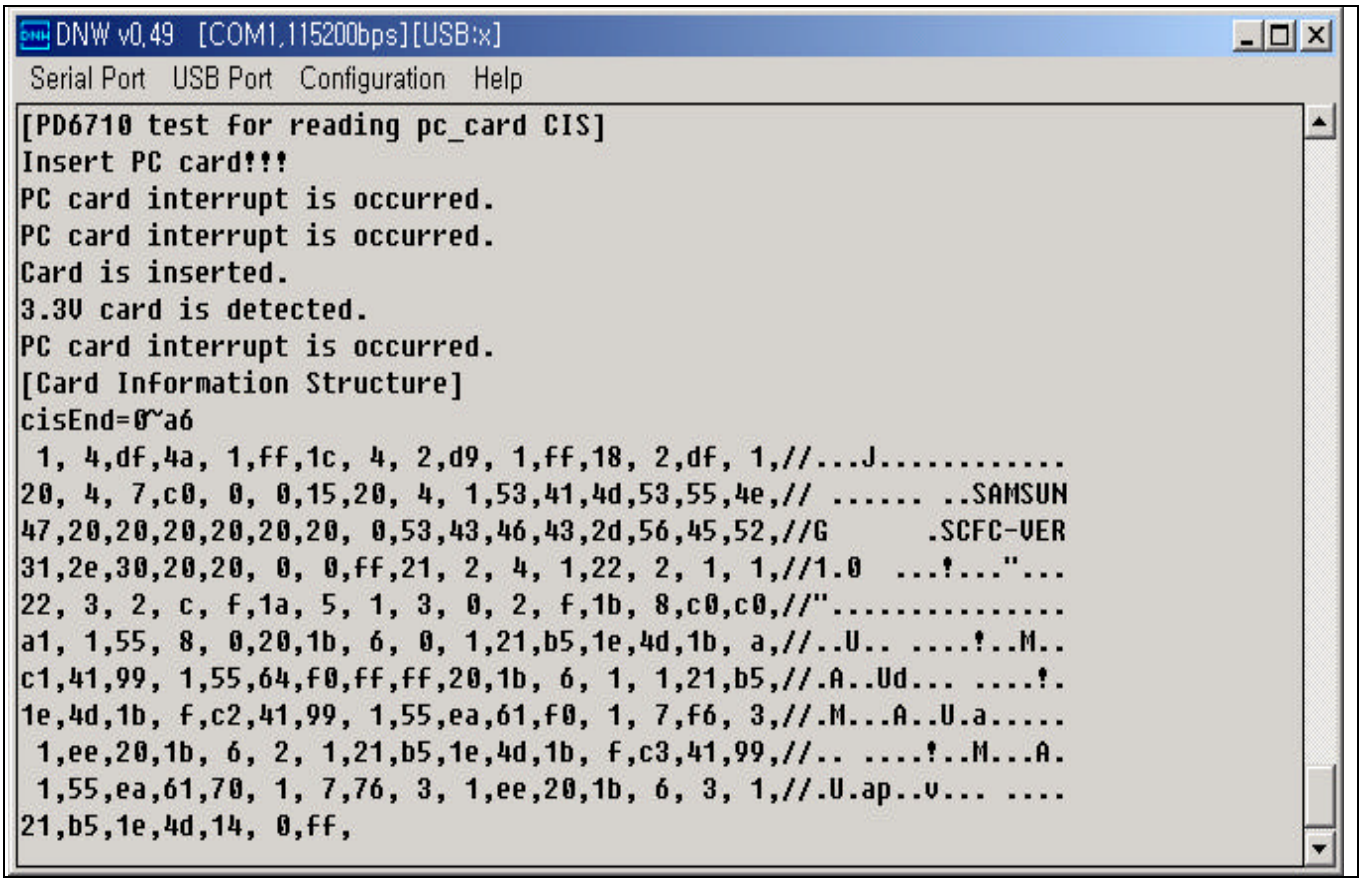


Figure 4-12. GPC Card CIS Access Example on S3C2410X

10BASE-T ETHERNET CONTROLLER (CS8900A) INTERFACE

The 10BASE-T Ethernet can be supported on S3C2410X using following components:

- CS-8900A from Cirrus logic
- XFMRS XF10B11A-COMB1-2S is Ethernet RJ45 with transformer.

AUDIO CODEC (UDA1341TS) CONNECTION WITH S3C2410X

The S3C2410X IIS interface example circuit is as follows:

- UDA1341TS from Philips Semiconductors.
- The L3 interface of Philips (L3MOD, L3CLOCK and L3DATA) is realized by general I/O port.
- Refer to the sample code of audio application which plays GPCM file.

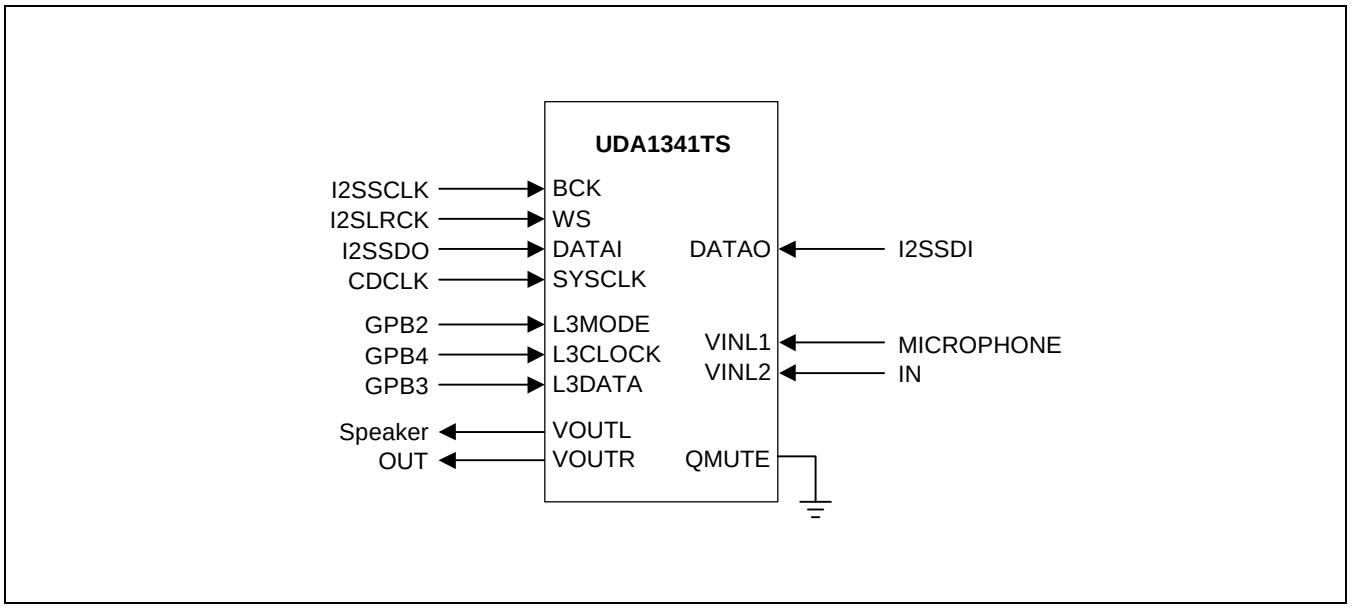


Figure 4-13. UDA1341TS Connection with S3C2410X

LCD CONNECTION WITH S3C2410X

The S3C2410X LCD interface example circuit is as follows:

- UG-32F04 (320x240 mono STN LCD) from SAMSUNG DISPLAY DEVICES CO., LTD. (refer to Figure 4-14)
  - TL497CAN can be used to make VEE (-25V).
- UG-24U03A (320x240 mono STN LCD) from SAMSUNG DISPLAY DEVICES CO., LTD. (refer to Figure 4-15)
  - VEE is generated by the circuit on LCD module.
  - VL is 2.4V typically.
  - DISPON H: display on, L: display off
  - nEL_ON H: EL off L: EL on
- KHS038AA1AA-G24 (256 color STN LCD) from KYOCERA Co. (refer to Figure 4-16)
  - DISP signal can be made using I/O port, or power control circuit or nRESET circuit.
  - V1-V5 can be made using the power circuit recommended by the LCD specification.
- LTS350Q1 (64K color TFT LCD) from SAMSUNG ELECTRONICS CO., LTD. (refer to Figure 4-17)
  - VDD_LCDI is typically 3.3V.
- LP104V2-W (262,144 color TFT LCD, 10.4") from LG Philips (refer to Figure 4-18)
  - VDD_LCDI is typically 3.3V.
- V16C6448AB (640x480 TFT LCD) from PRIMEVIEW (refer to Figure 4-19)
  - VDD_LCDI, VD and control signal are typically 5.V.

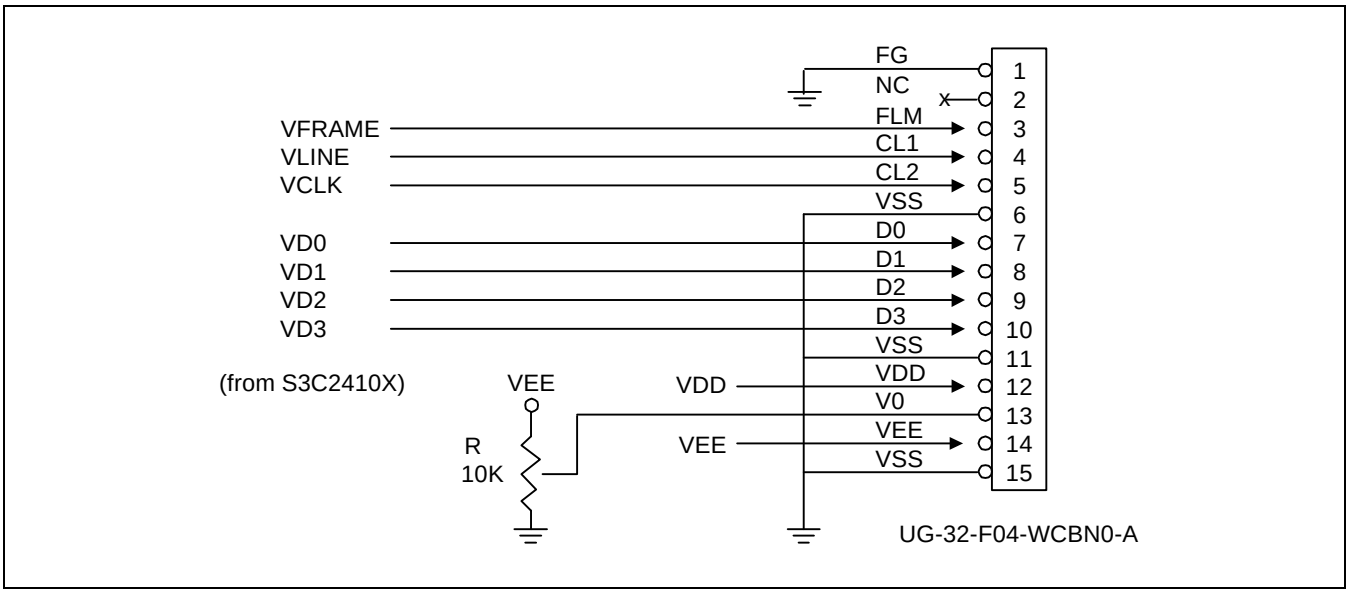


Figure 4-14. UG-32F04 Connection with S3C2410X (320x240 Mono STN LCD)

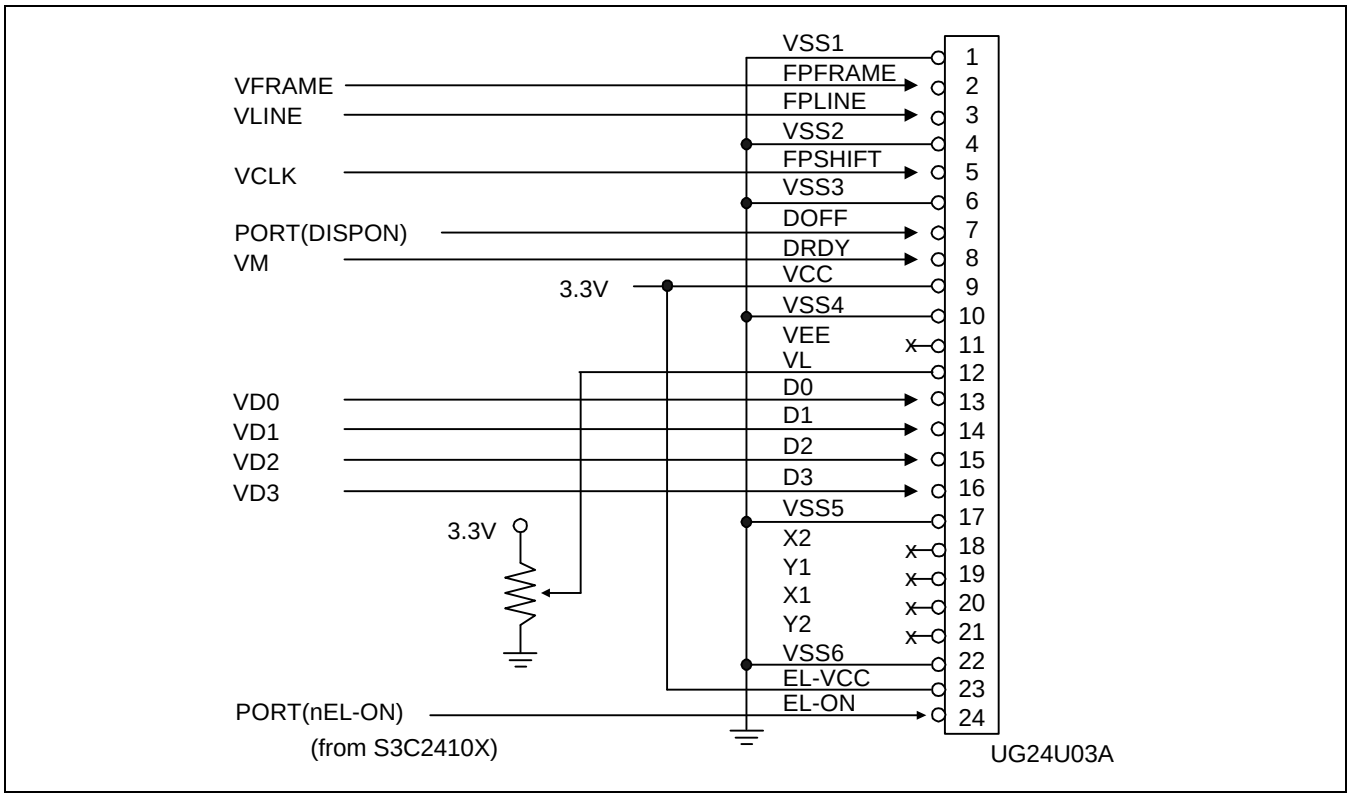


Figure 4-15. UG24U03A Connection with S3C2410X (320x240 Mono STN LCD)

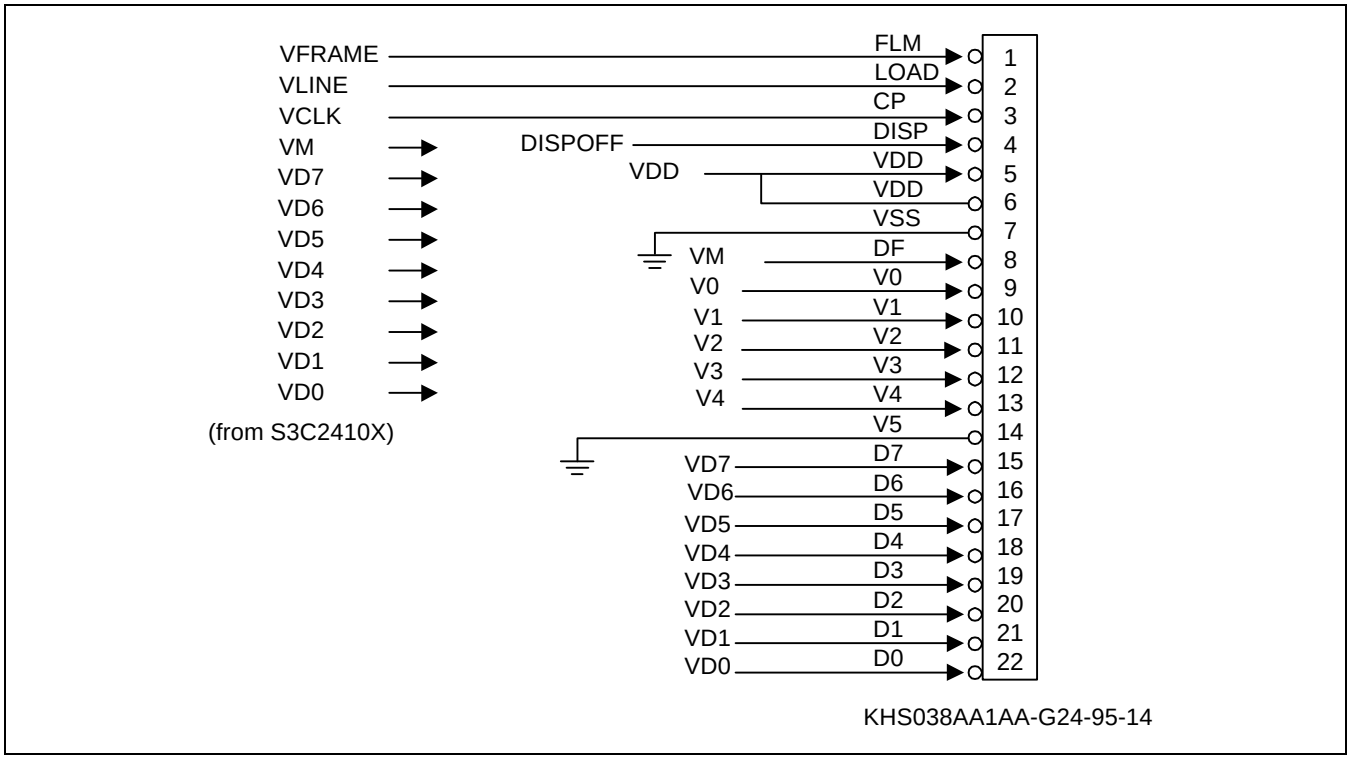


Figure 4-16. KHS038AA1AA-G24 Connection with S3C2410X (256 Color STN LCD)

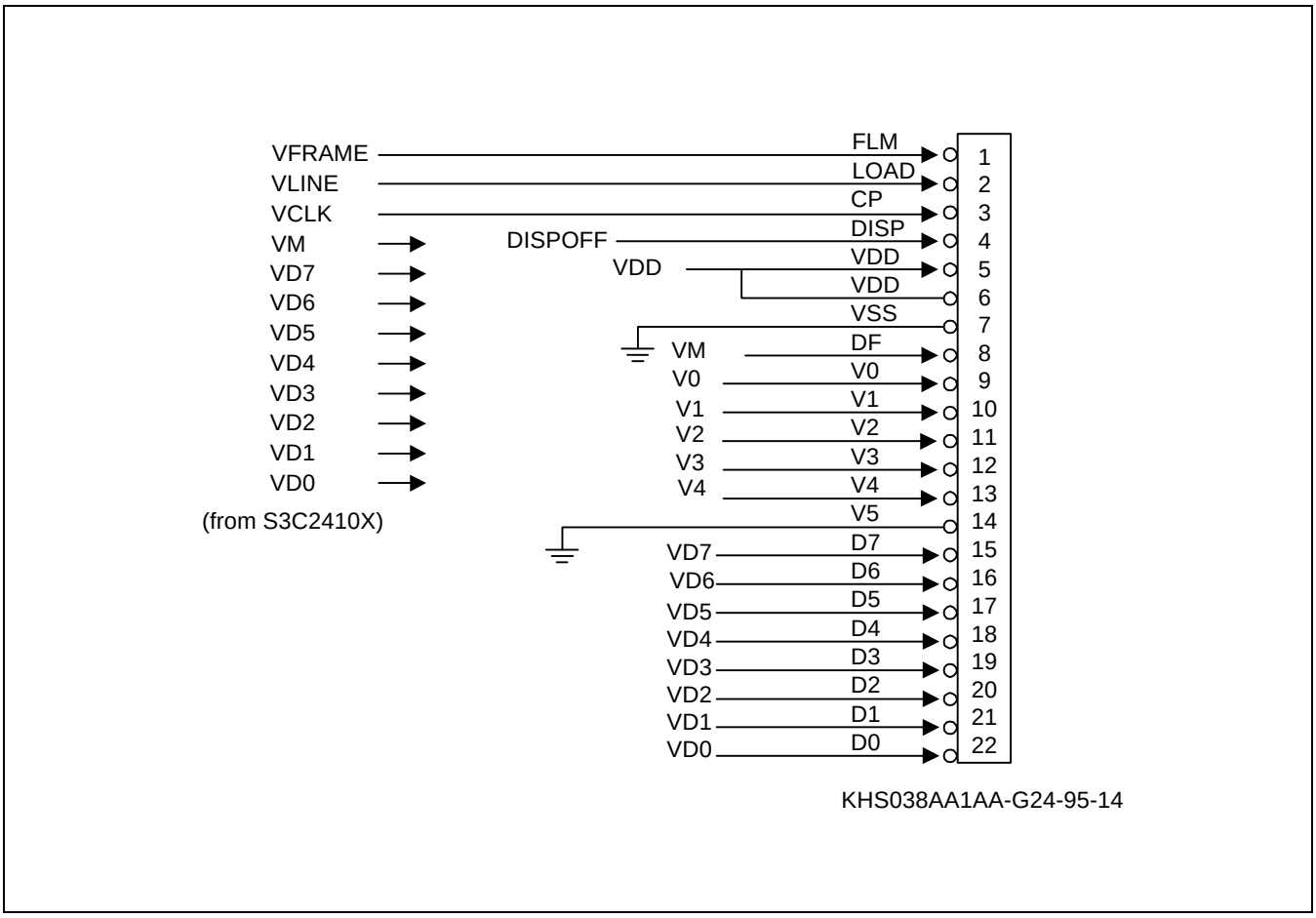


Figure 4-17. LTS350Q1 Connection with S3C2410X (Samsung 3.5” Reflective TFT LCD)

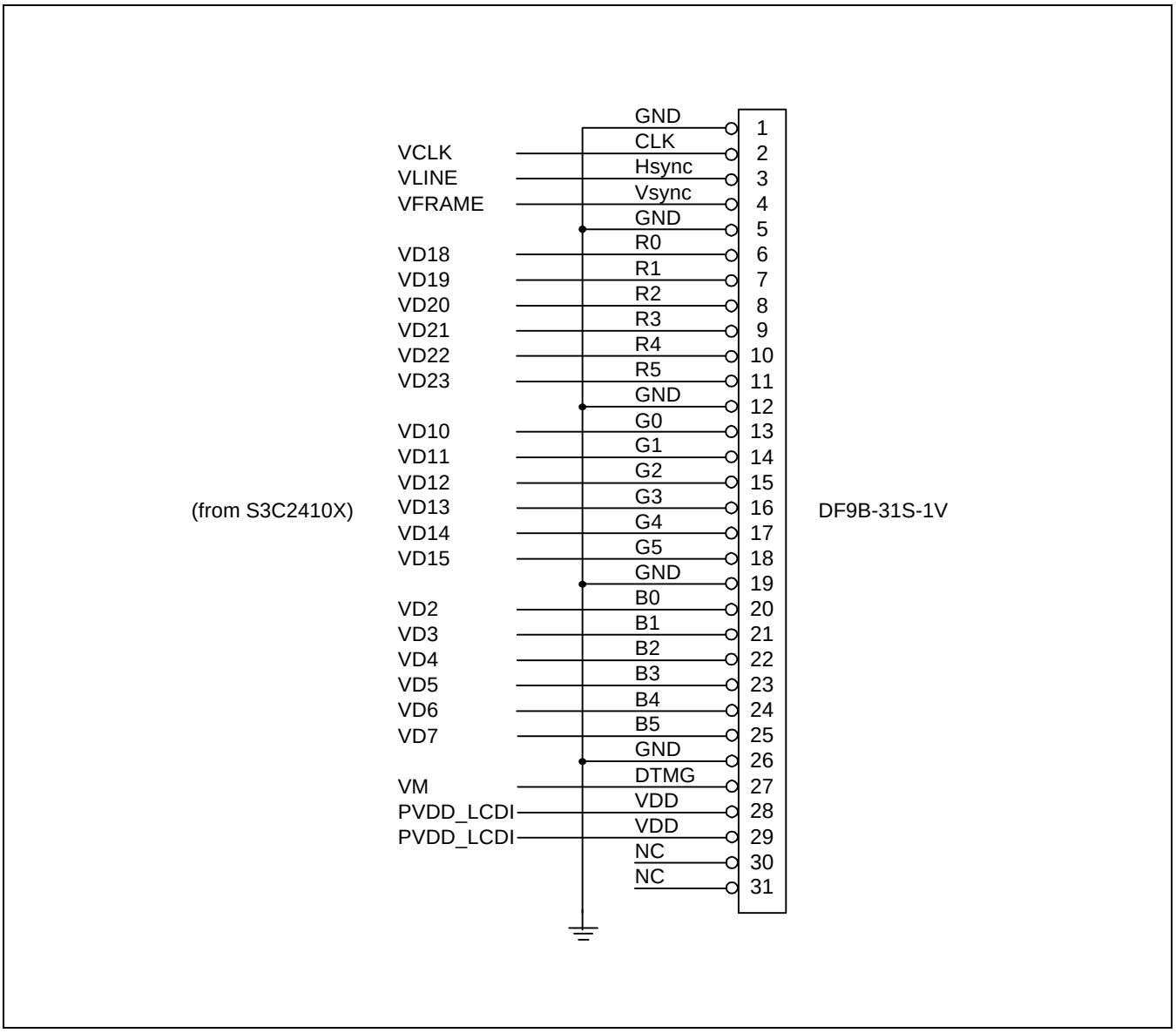


Figure 4-18. LP104V2-W Connection with S3C2410X (LG Philips 10.4” TFT LCD)

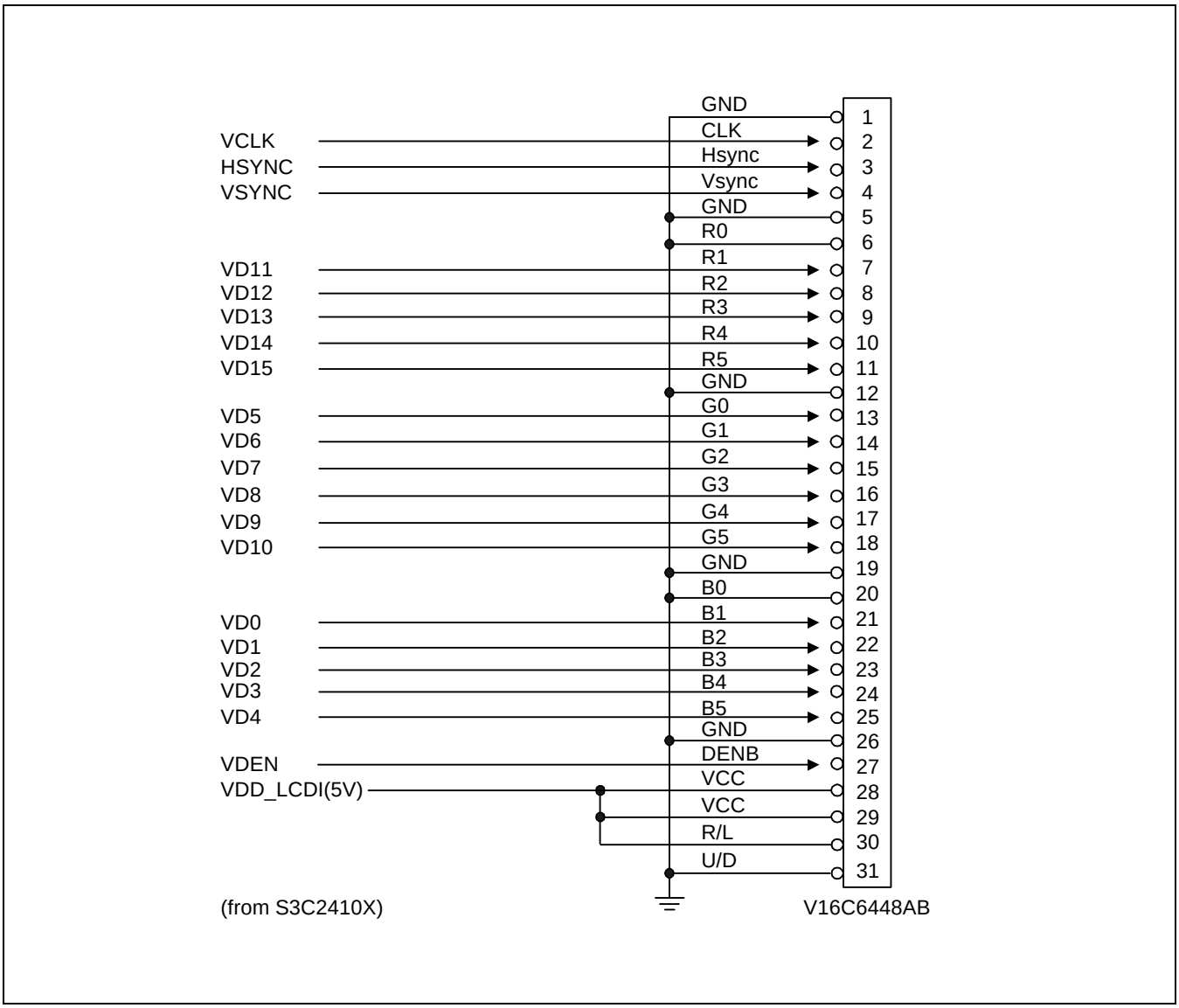


Figure 4-19. V16C6448AB Connection with S3C2410X (TFT LCD)

TOUCH SCREEN PANEL (TSP) INTERFACE CIRCUIT WSSITH S3C2410X

Typically, the TSP consists of two plane resisters (x-axis plane resister and y-axis plane resister). In this reason, TSP has four terminals. To read X coordinate, Q1 and Q2 are turned on while Q3 and Q4 is turned off. Therefore, X coordinate value can be read out from AIN7 by ADC. To read Y coordinates, Q3 and Q4 is turned on while Q1 and Q2 is turned off. So, ADC can read out Y coordinate value from AIN5. The INT_TC can be used to check whether the TSP is touched.

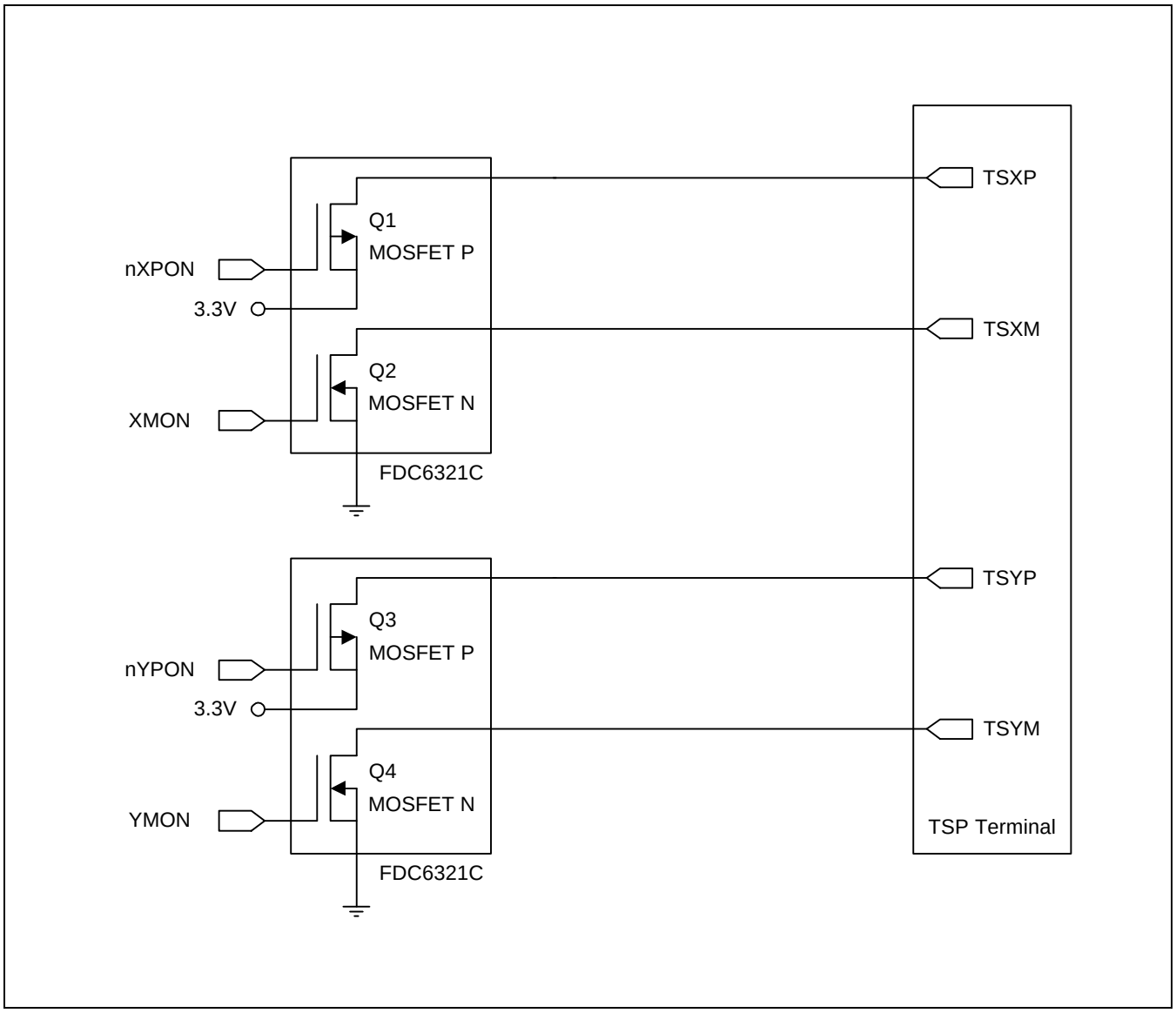


Figure 4-20. TSP Interface Circuit with S3C2410X

SYSTEM DESIGN WITH DEBUGGER SUPPORT

MULTI-ICE

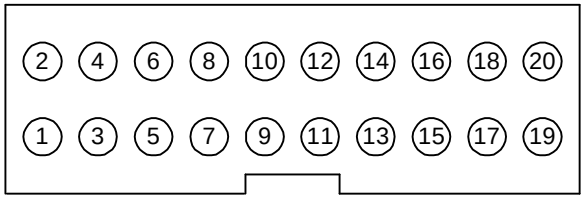
The S3C2410X has an EmbeddedICE logic that provides debug solution from ARM. MULTI-ICE enables you to debug software running on the S3C2410X. EmbeddedICE logic is accessed through the Test Access Port (TAP) controller on the S3C2410X using the JTAG interface.

JTAG port for EmbeddedICE Interface

When you build a system with the S3C2410X EmbeddedICE interface, you should design a JTAG port for MULTI-ICE interface. Usually, the interface connector is a 20-way box header, and this plug is connected to the EmbeddedICE logic interface module using 20-way IDC socket.

The JTAG port signals, nTRST, TDI, TMS and TCK have to be connected to pulled-up register (10K ohm) externally.

The pin configuration and a sample design are described in Figure 4-21 and Figure 4-22, respectively.



Pin	Name	Function
1	VTref	System Power
2	Vsupply	System Power
3	nTRST	Test reset, active low (connected pull-up reg.)
5	TDI	Test data in (connected pull-up reg.)
7	TMS	Test mode select (connected pull-up reg.)
9	TCK	Test clock (connected pull-up reg.)
11	RTCK	Return test clock (connected pull-down reg.)
13	TDO	Test data out
15	nSRST	Connected to nRESET and nTRST through 470 ohm resister
17	DBGRQ	NC
19	DBGACK	NC
4, 6, 8, 10, 12, 14, 16, 18, 20	GND	System Ground

Figure 4-21. MULTI-ICE Interface of JTAG Connector

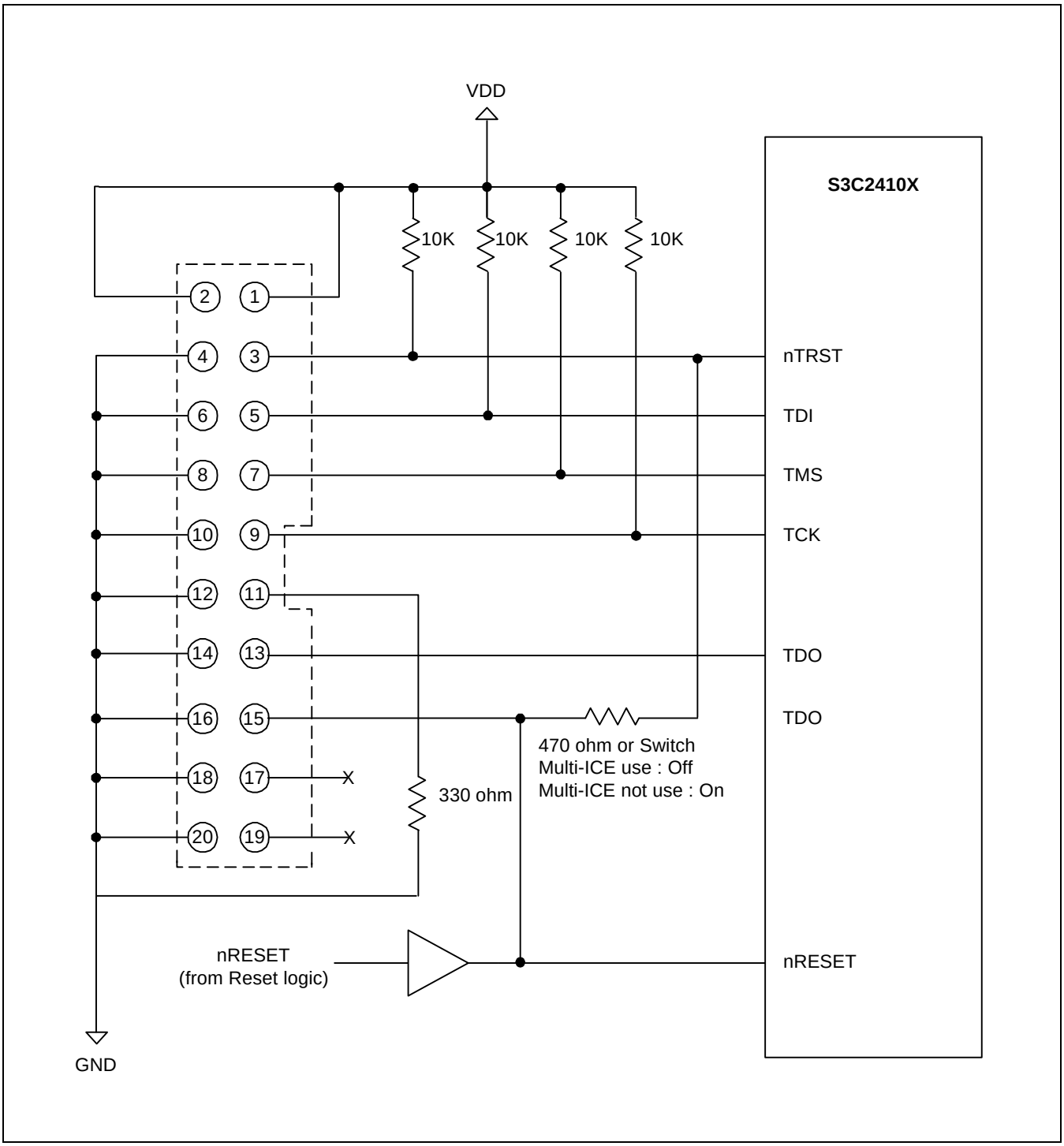


Figure 4-22. MULTI-ICE Interface Design Example

CHECK ITEMS FOR SYSTEM DESIGN WITH S3C2410X

When you design a system with the S3C2410X, you should check a number of items to build a good system. The check items are described below.

- The OM[3:0] pin has to be configured.
- If EXTCLK pin is used for MPLL and UPLL, XTIpll has to be connected to VDD. If XTIpll pin is used for MPLL and UPLL, EXTCLK has to be connected to VDD.
- If an input pin is unused, connect the pin to VDD or GND. If the pin is floated, S3C2410X may not operate.

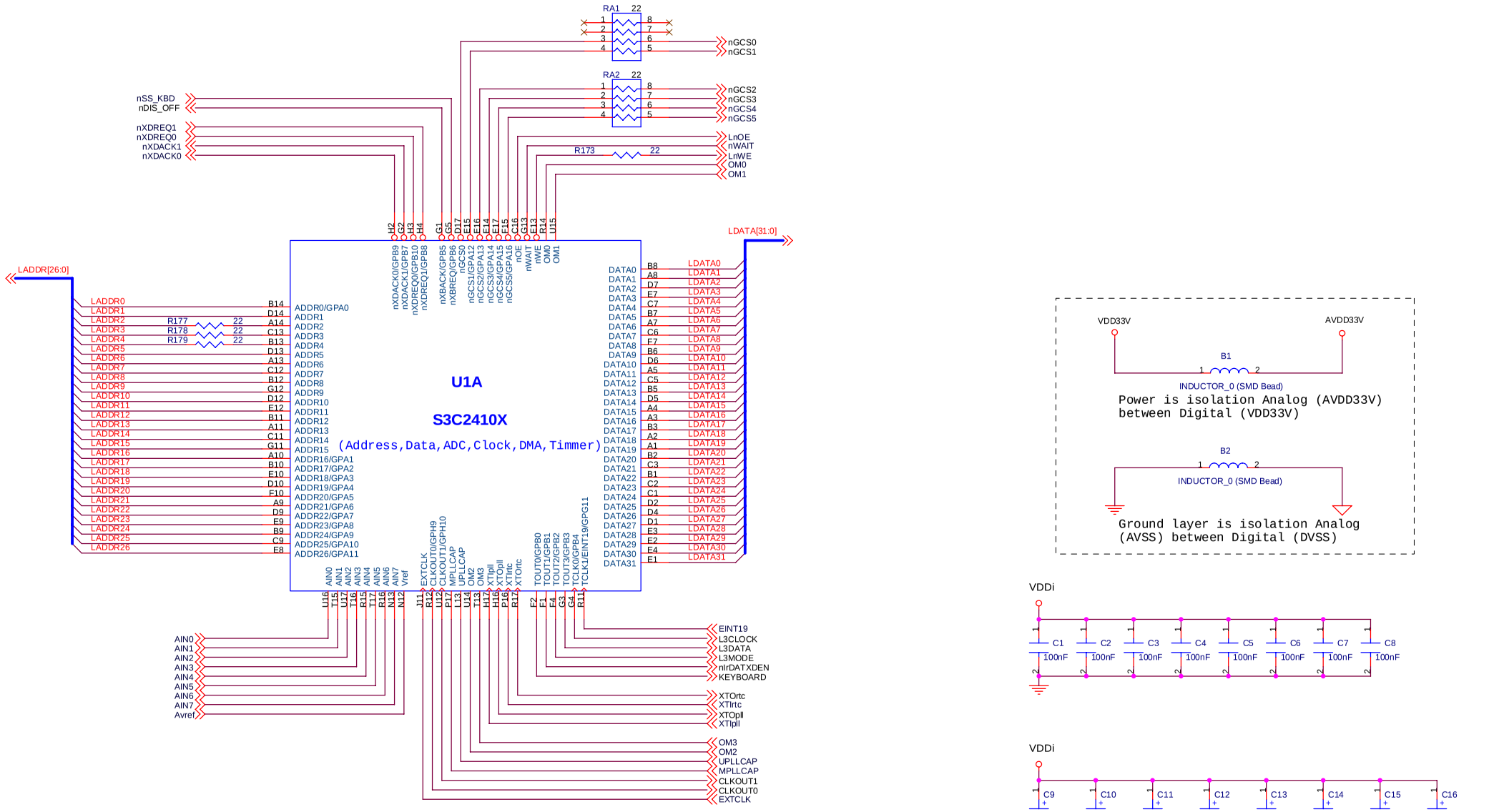
NOTES

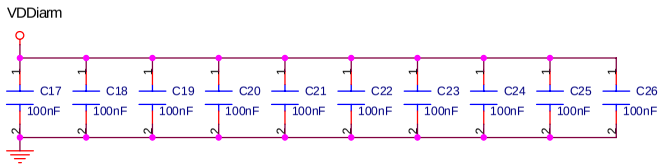
SMDK2410 Board (S3C2410 Reference Board)

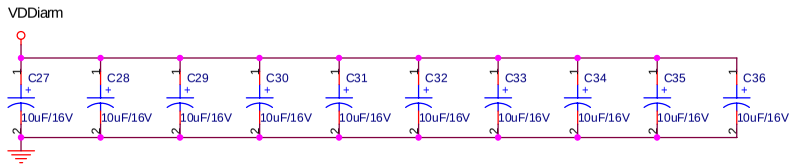
PCB Revision	Date	Description
Ver : 1.0 (20020417)	April 17, 2002	First release version (Production for Customer)
Ver : 1.1 (20020509)	May 09, 2002	1. Sheet 2 (Coordinate B5) , 3 (B4) , 10 (D5) : SD card(CON11) function add CD(Card Detect) and WP(Write Protect) -> Add R169 (Unload), R170 (0), R171 (4.7K) 2. Sheet 1 (Coordinate D3, C5) , 2 (C5) : Add damping resister to SDRAM signal.(LSCLK0 10 ohm, LnWE/LnGCS6/LnSCAS/LnSRAS/LADDR2,3,4 22 ohm : R172 10 ohm , R173 ~ R179 22 ohm)
Ver : 1.2 (20020603)	June 03, 2002	1. Sheet 2 (Coordinate C5) : U1B-F14 is changed from TP2 (Probe SCLK1) to LSCLK1 signal -> Add R180 (10 ohm) 2. Sheet 4 (Coordinate C3) : U9-38 is changed from LSCLK0 signal to LSCLK1 signal 3. Sheet 8 (Coordinate A4,B4) : R167, R168 is changed from 0 ohm to 4.7K, and C187,C188 is changed from unload to 1nF.
Ver : 1.3 (20020703)	July 03, 2002	1. Sheet 2,5 (Coordinate B5,C4) : Changed, J2 2Pin ---> 3Pin 2. Sheet 10 (Coordinate D5) : Deleted, R106, R107, R109, R113 ~ R115 (0 ohm) 3. Sheet 10 (Coordinate D5) : Added, R181 ~ R186 (10Kohm)

Title	SMDK2410X (S3C2410X DEMO BOARD)	
Size A3	Document Number Revision History	Rev 1.3
Date: Monday, August 19, 2002	Sheet 0 of 12	

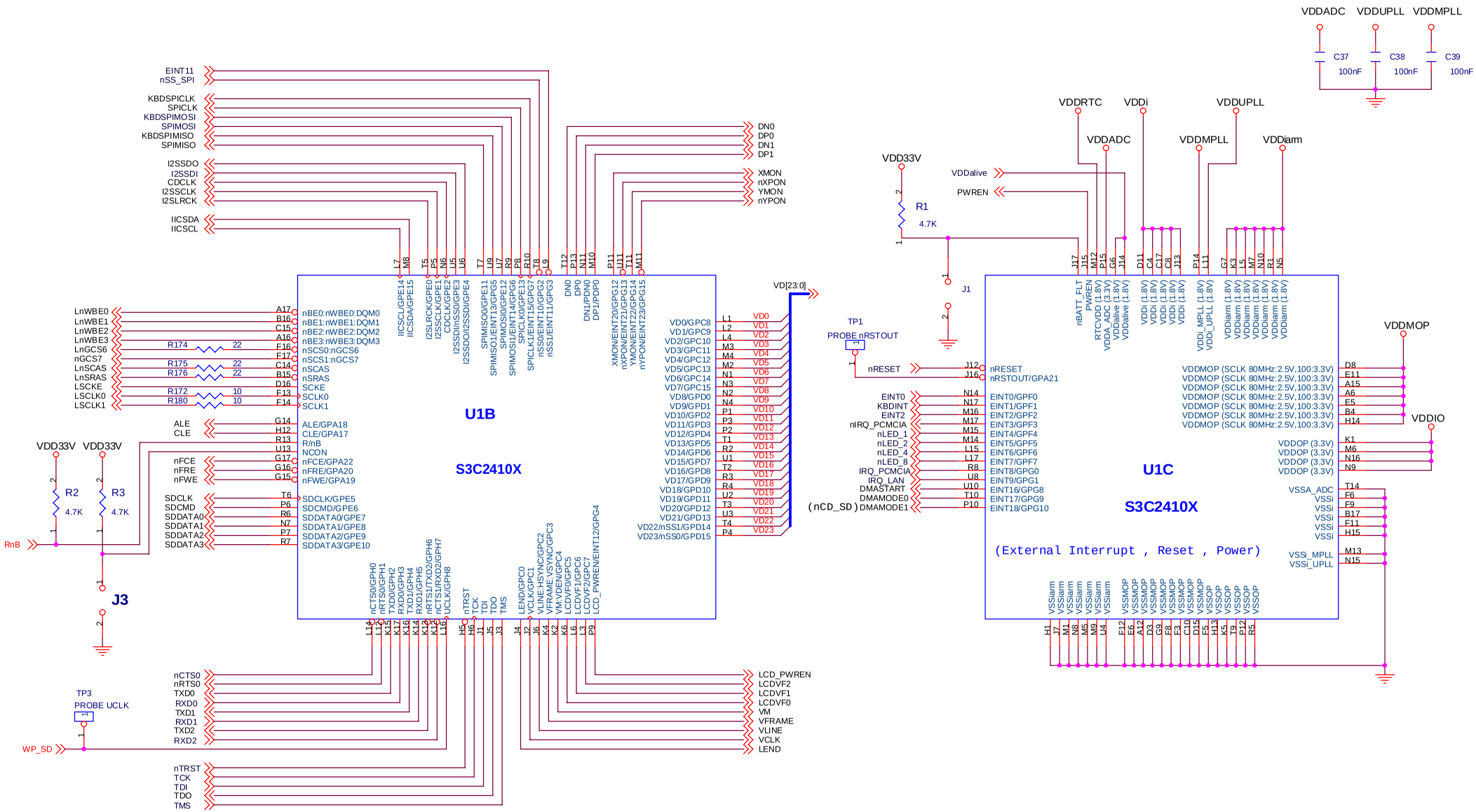
NOTE!!!
Lxxxx signals should be routed as short as possible.





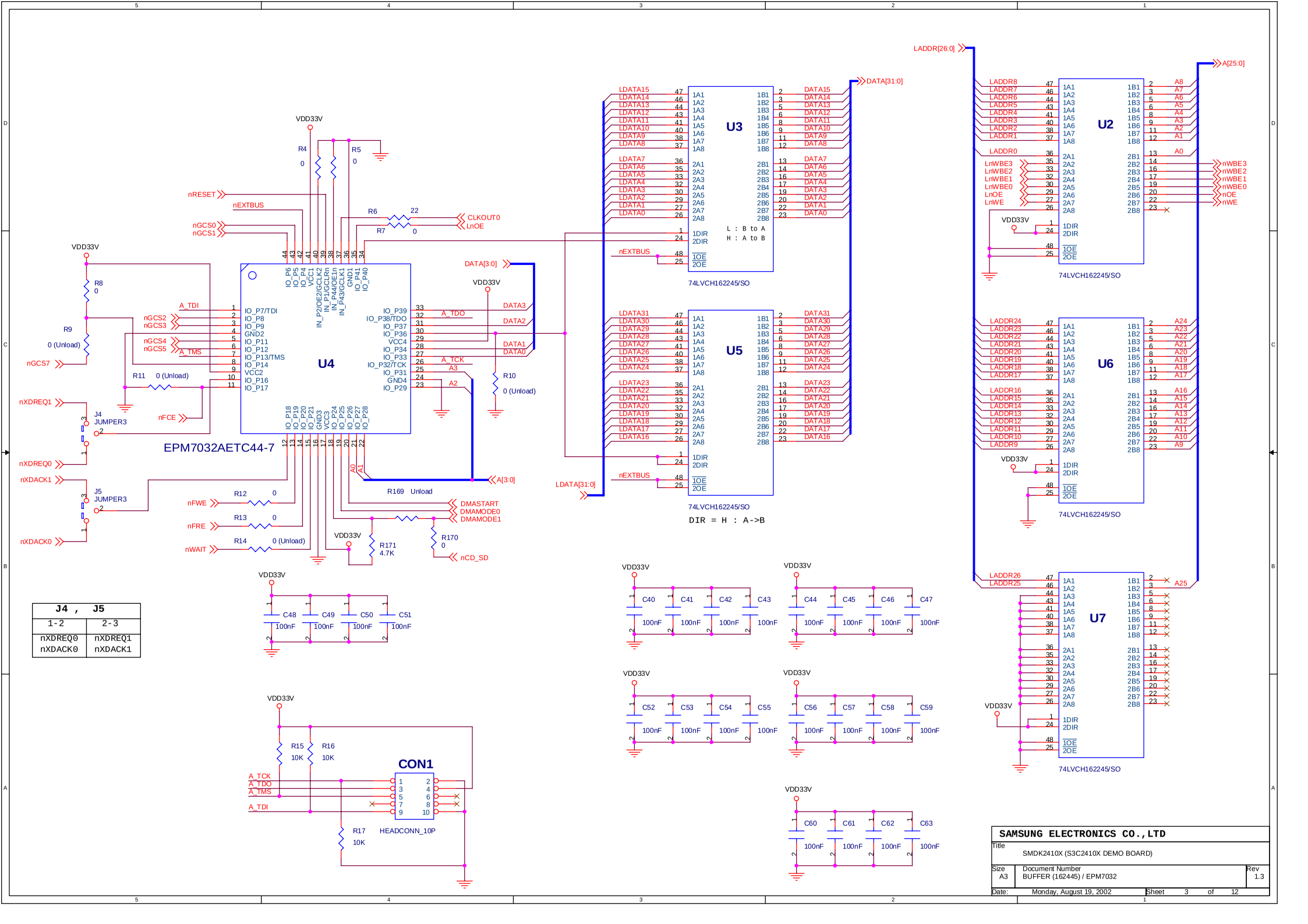


SAMSUNG ELECTRONICS CO.,LTD		
Title: SMDK2410X (S3C2410X DEMO BOARD)		
Size: A3	Document Number: MCU (ADDR/DATA/ADC/CLOCK/TIMMER/INT/RESET/POWER)	Rev: 1.3
Date: Monday, September 16, 2002	Sheet 1 of 12	



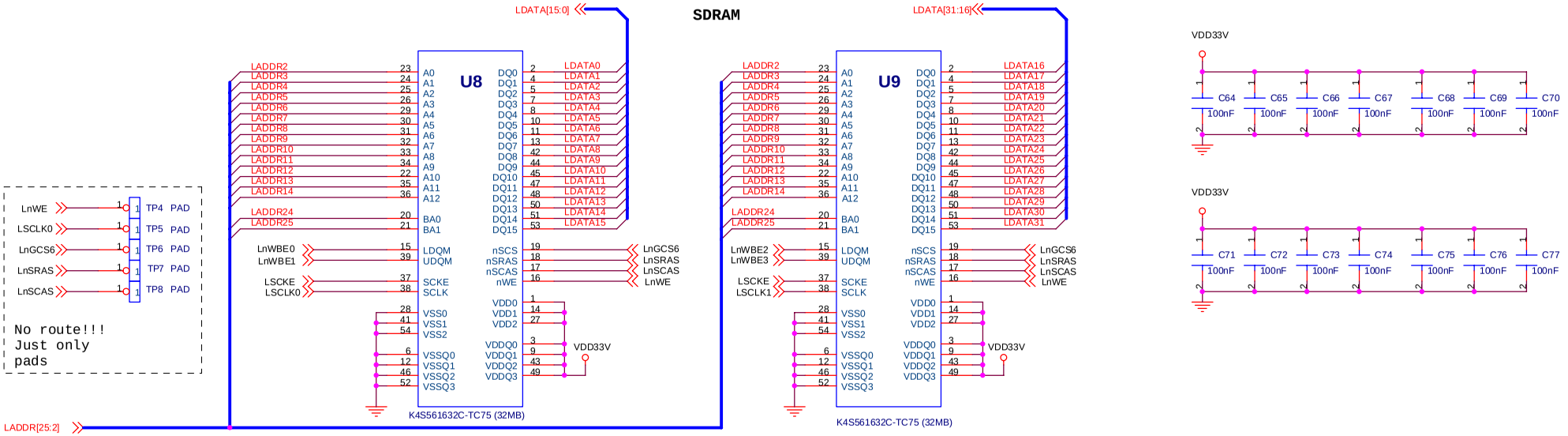
Address Step J3 (NCON)	NAND Flash Configuration
0	512 Byte/Page , 3 Step Addressing
1	512 Byte/Page , 4 Step Addressing

SAMSUNG ELECTRONICS CO.,LTD		
Title: SMDK2410X (S3C2410X DEMO BOARD)		
Size: A3	Document Number: MCU (LCD/JTAG/UART/USB/NAND/MMC/UART/USB/SPI/MEMORY)	Rev: 1.3
Date: Monday, September 16, 2002	Sheet 2 of 12	



SAMSUNG ELECTRONICS CO.,LTD		
Title: SMDK2410X (S3C2410X DEMO BOARD)		
Size: A3	Document Number: BUFFER (162445) / EPM7032	Rev: 1.3
Date: Monday, August 19, 2002	Sheet 3 of 12	

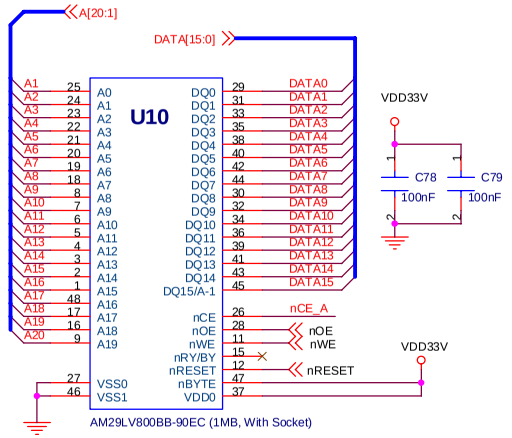
SDRAM



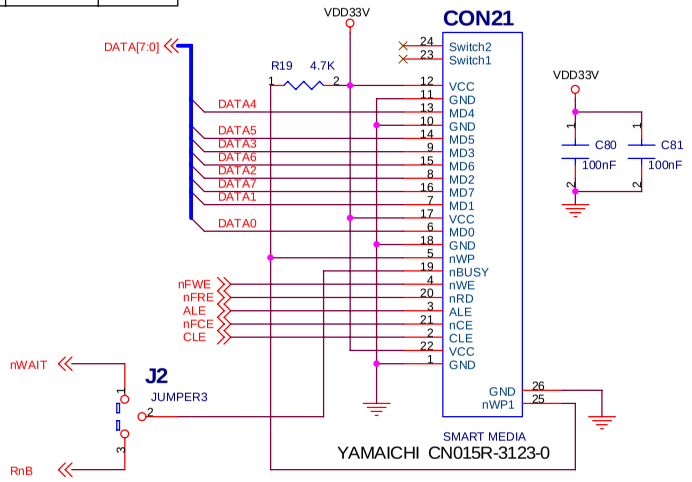
SAMSUNG ELECTRONICS CO.,LTD		
Title: SMDK2410X (S3C2410X DEMO BOARD)		
Size: A3	Document Number: SDRAM	Rev: 1.3
Date: Monday, August 19, 2002	Sheet 4 of 12	

S3C2410X01	J2	
Revision 0	1 - 2	nWAIT
Revision 1	2 - 3	RnB

AMD Flash Memory(SOCKET)

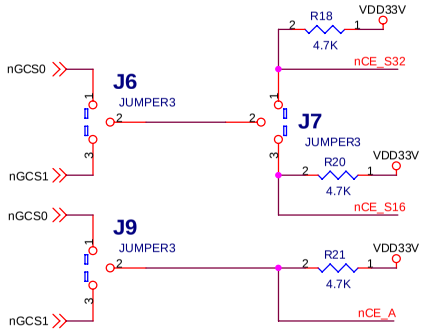


SmartMedia Card (NAND Flash memory)

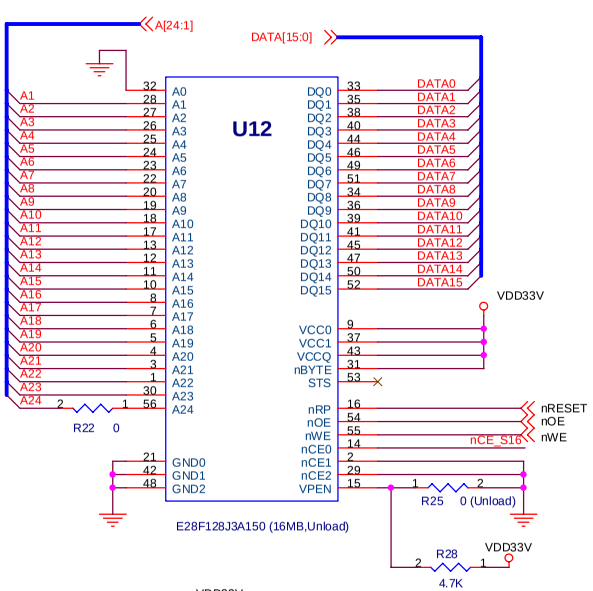


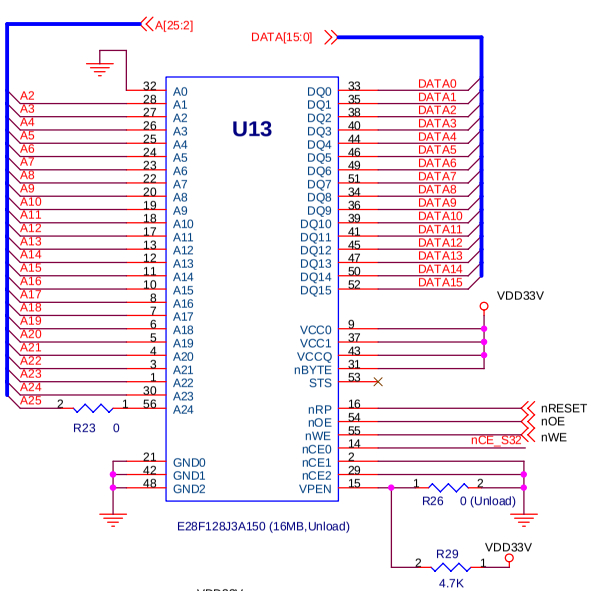
FLASH Memory Select

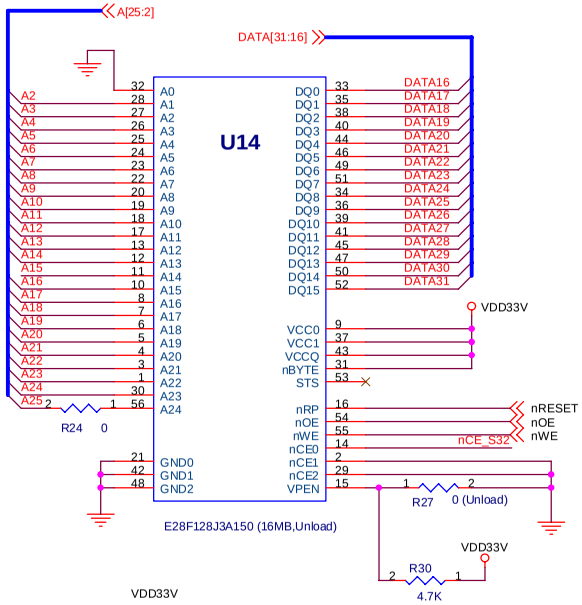
FLASH	J6	J9	J7
AMD Flash	2-3	1-2	1-2
Strata 32bit	1-2	2-3	1-2
Strata 16bit	1-2	2-3	2-3

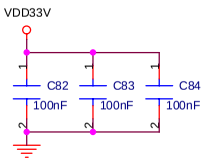


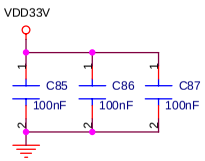
INTEL StrataFlash Memory(SMD Unload)

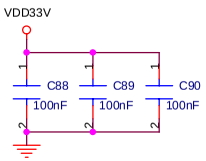




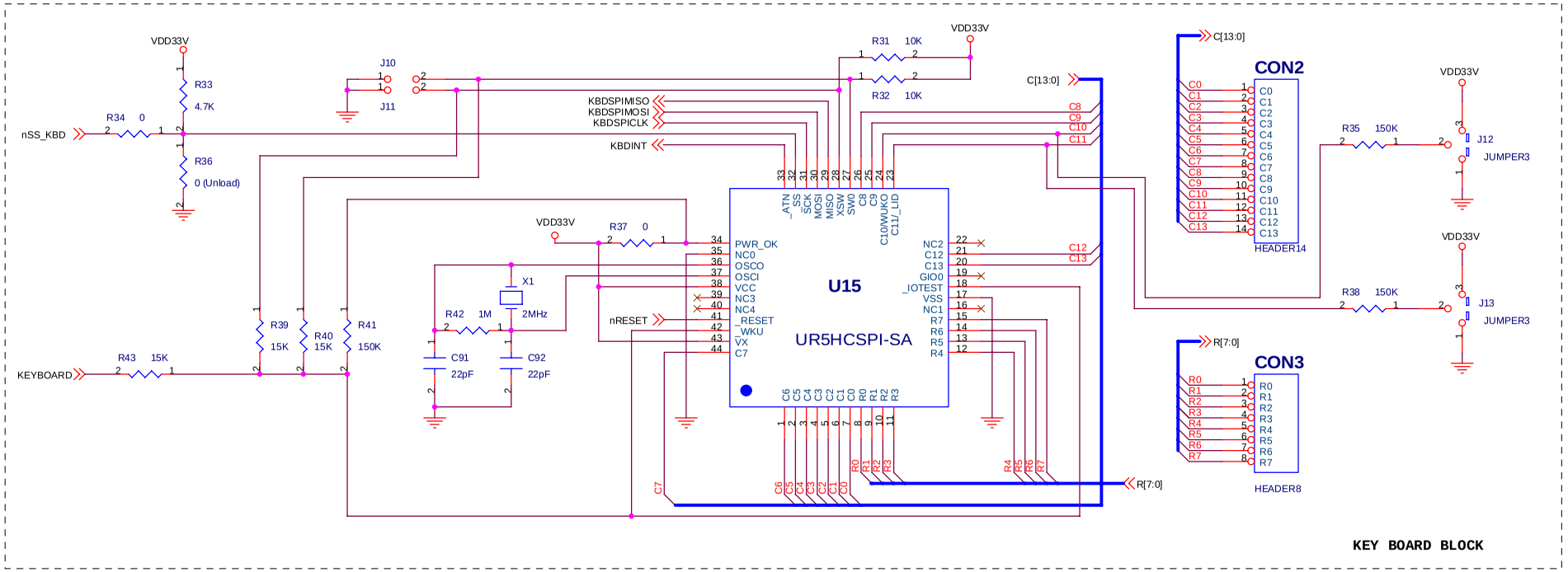


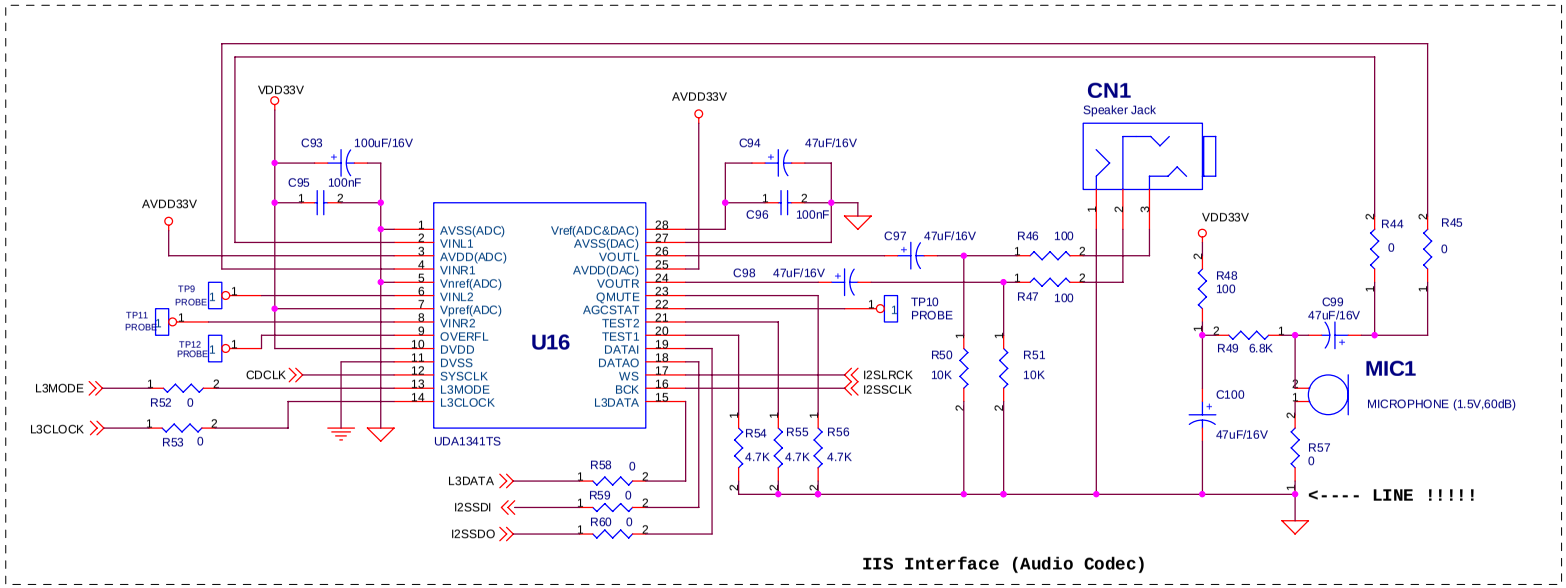




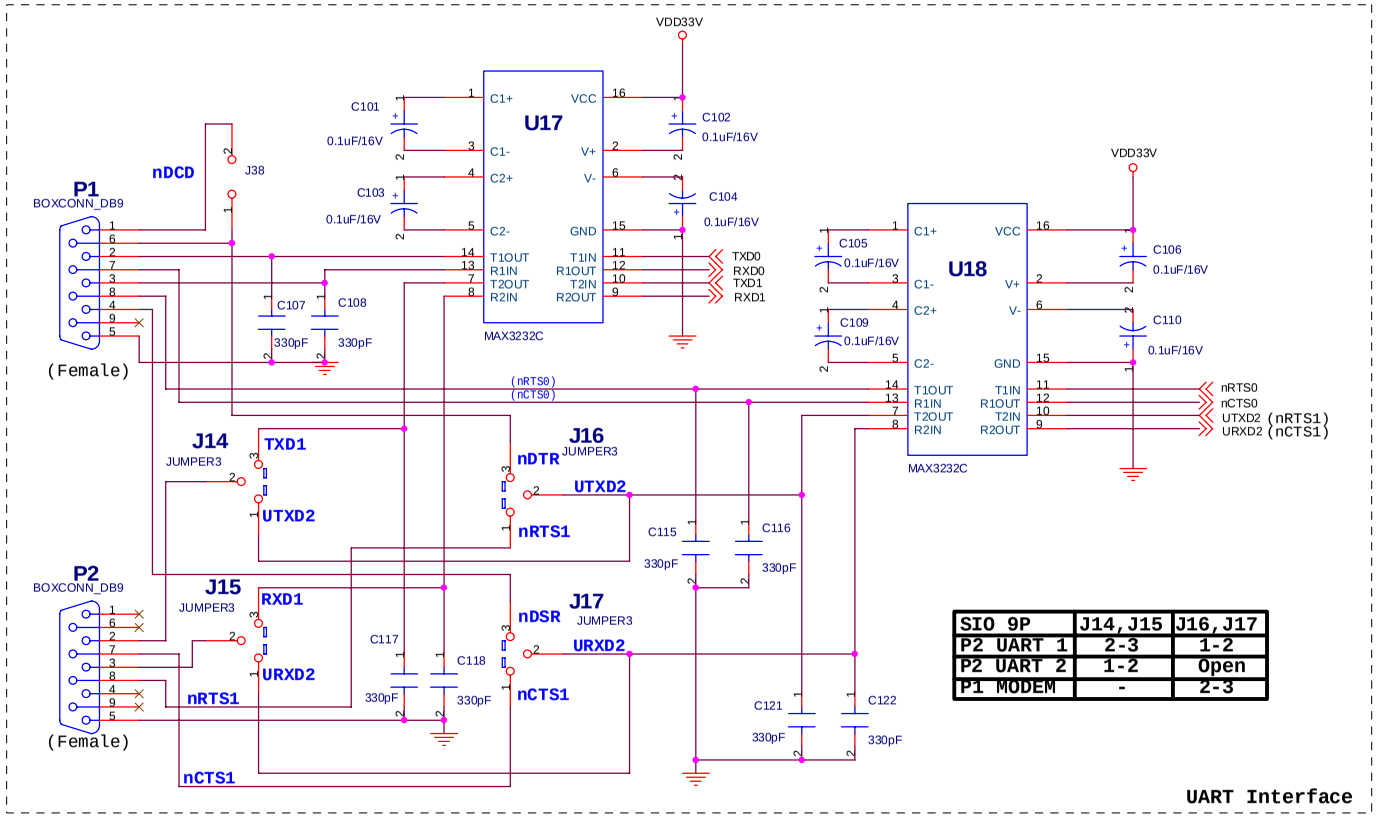


SAMSUNG ELECTRONICS CO.,LTD		
Title: SMDK2410X (S3C2410X DEMO BOARD)		
Size: A3	Document Number: AMD & NOR(INTEL) , NAND(SAMSUNG) FLASH MEMORY	Rev: 1.3
Date: Monday, August 19, 2002	Sheet 5 of 12	



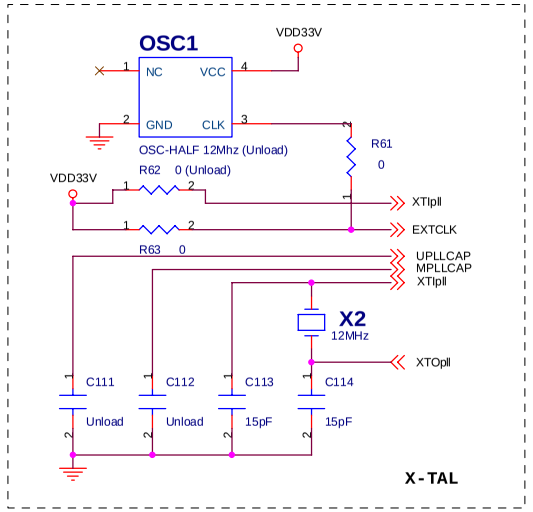


SAMSUNG ELECTRONICS CO.,LTD		
Title: SMDK2410X (S3C2410X DEMO BOARD)		
Size: A3	Document Number: KEYBOARD / IIS (AUDIO CODEC)	Rev: 1.3
Date: Monday, August 19, 2002	Sheet 6 of 12	

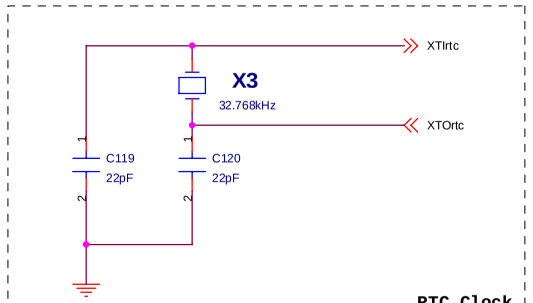


SIO 9P	J14,J15	J16,J17
P2 UART 1	2-3	1-2
P2 UART 2	1-2	Open
P1 MODEM	-	2-3

UART Interface

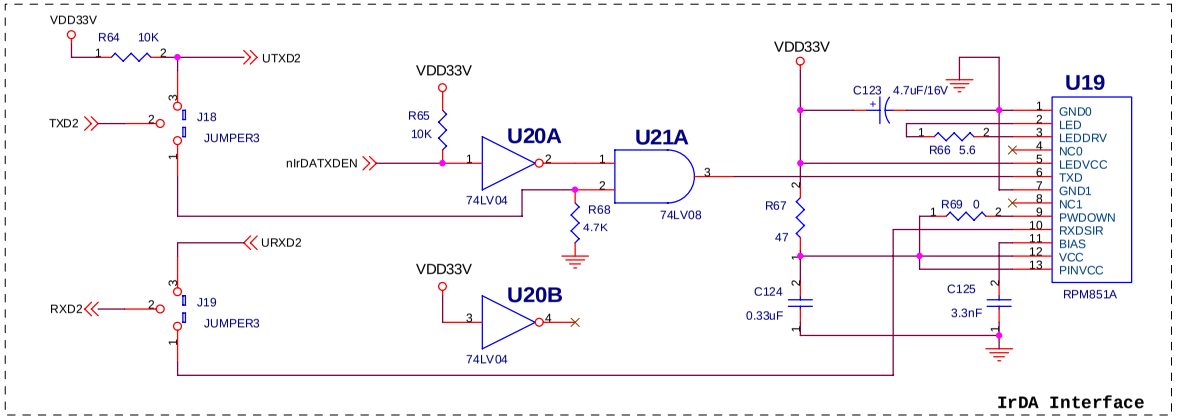


X-TAL

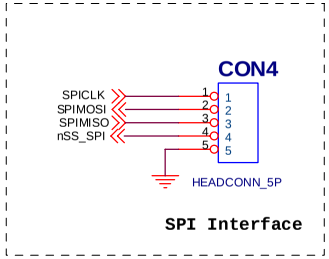


RTC Clock

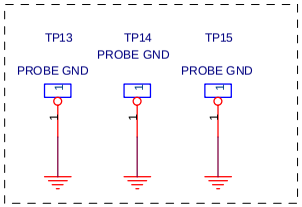
	J18	J19
IrDA	1-2	1-2
UART2	2-3	2-3



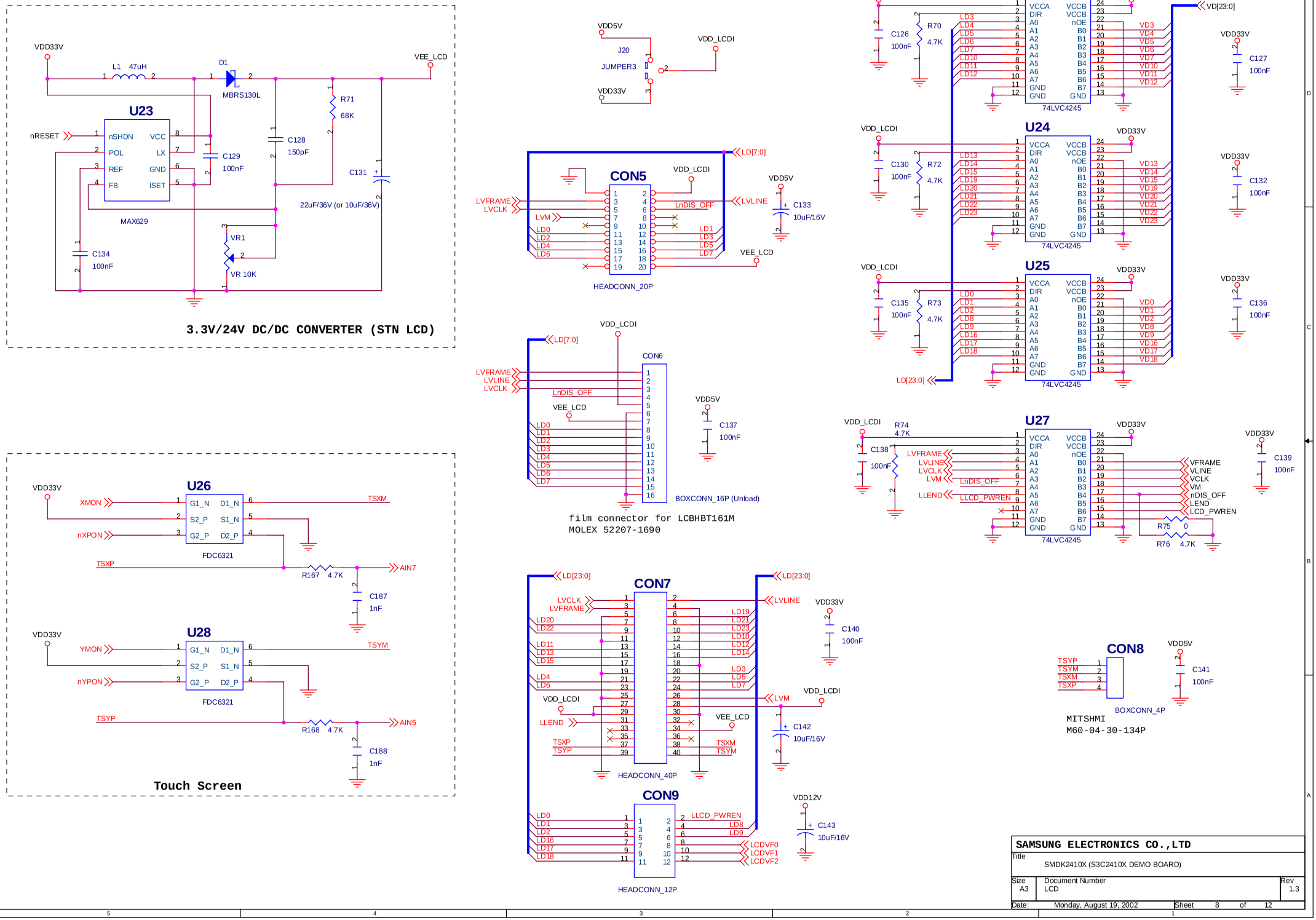
IrDA Interface

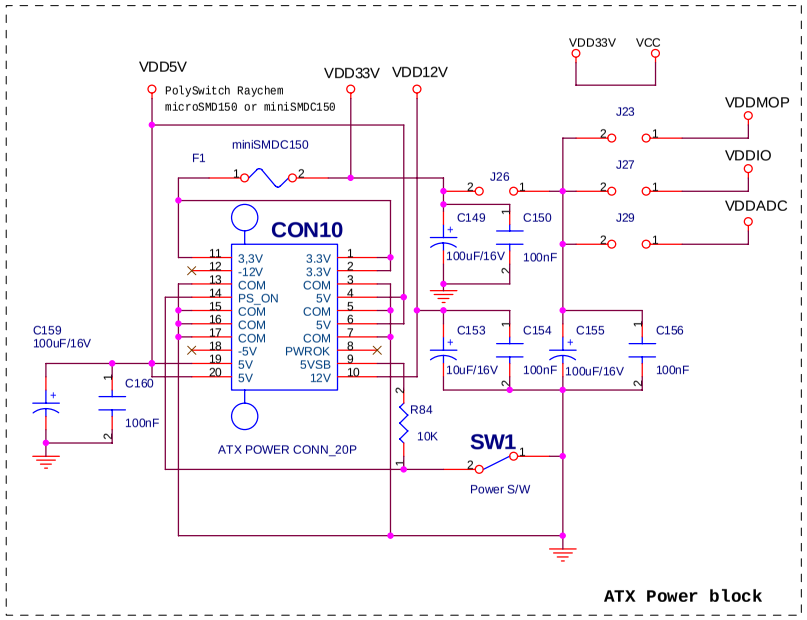


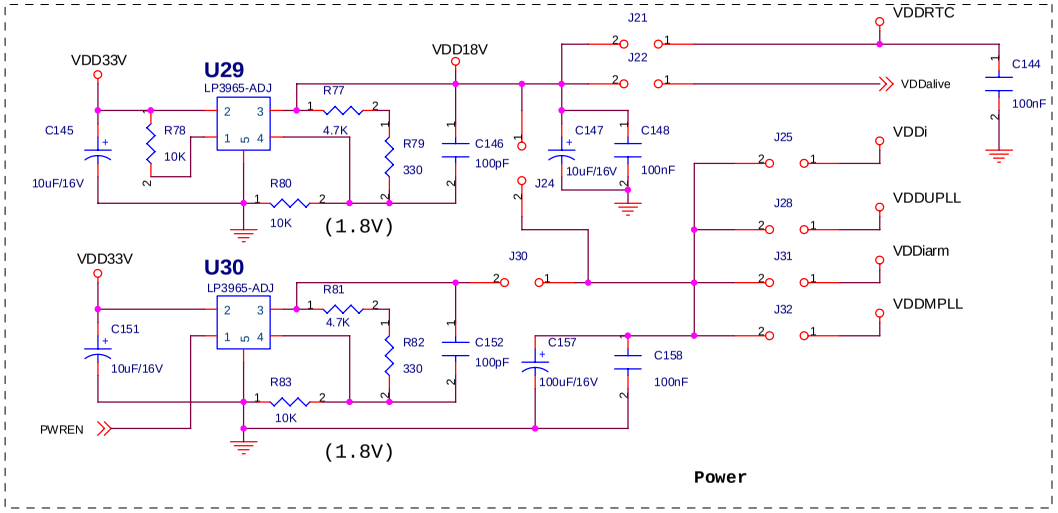
SPI Interface

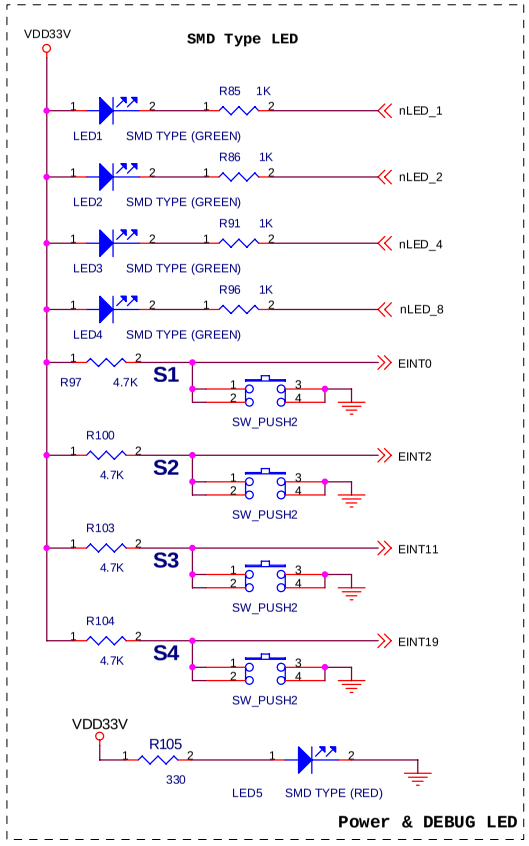


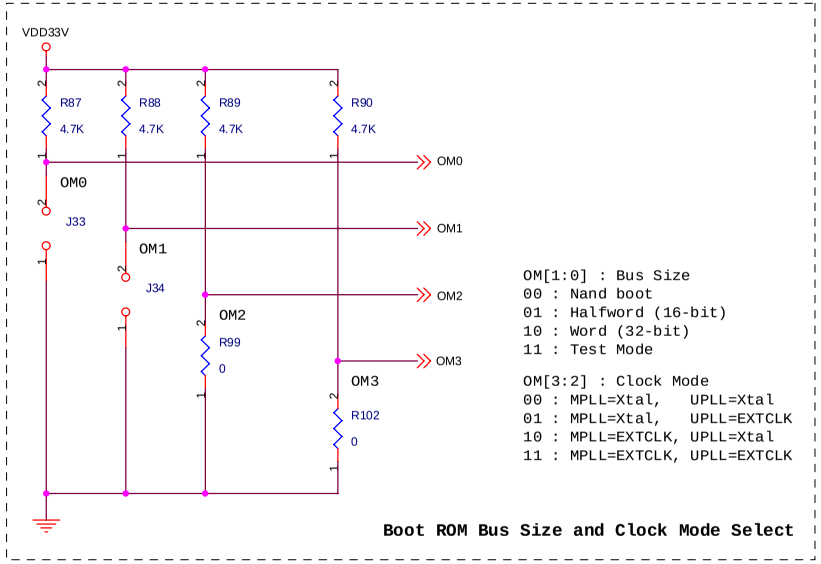
SAMSUNG ELECTRONICS CO.,LTD	
Title	SMDK2410X (S3C2410X DEMO BOARD)
Size: A3	Document Number: UART / Clock / IrDA / SPI — Rev: 1.3
Date: Monday, August 19, 2002	Sheet 7 of 12

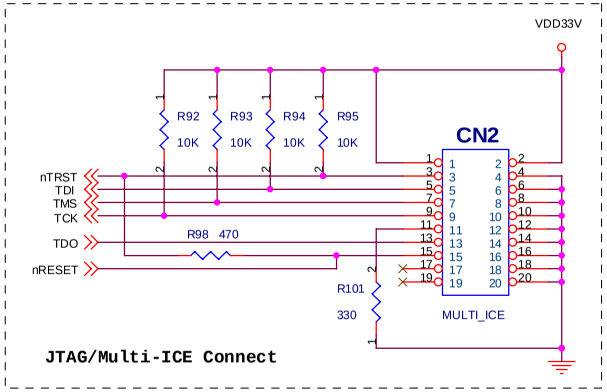

VDD33V
L1 47uH
D1
MBRS130L
VEE_LCD
U23
nRESET
nSHDN
VCC
POL
LX
REF
GND
FB
ISET
MAX629
C129
100nF
C128
150pF
R71
68K
C131
22uF/36V (or 10uF/36V)
C134
100nF
VR1
VR 10K
3.3V/24V DC/DC CONVERTER (STN LCD)
VDD5V
J20
JUMPER3
VDD33V
VDD_LCDI
CON5
LD[7:0]
LVFRAME
LVCLK
LVM
LD0
LD2
LD4
LD6
LnDIS_OFF
LD1
LD3
LD5
LD7
LVLINE
C133
10uF/16V
VEE_LCD
HEADCONN_20P
CON6
LVLINE
VEE_LCD
C137
100nF
BOXCONN_16P (Unload)
film connector for LCBHBT161M
MOLEX 52207-1690
U22
U24
U25
U27
VCCA
VCCB
DIR
A0
nOE
A1
B0
A2
B1
A3
B2
A4
B3
A5
B4
A6
B5
A7
B6
GND
B7
74LVC4245
VD[23:0]
LD[23:0]
C126
100nF
R70
4.7K
C127
100nF
C130
100nF
R72
4.7K
C132
100nF
C135
100nF
R73
4.7K
C136
100nF
R74
4.7K
C138
100nF
VFRAME
VLINE
VCLK
VM
nDIS_OFF
LEND
LCD_PWREN
R75 0
R76 4.7K
C139
100nF
U26
U28
XMON
nXPON
YMON
nYPON
G1_N
D1_N
S2_P
S1_N
G2_P
D2_P
FDC6321
TSXM
TSYM
TSXP
TSYP
R167 4.7K
R168 4.7K
AIN7
AIN5
C187
1nF
C188
1nF
Touch Screen
CON7
LVCLK
LVFRAME
LD20
LD22
LD11
LD13
LD15
LD4
LD6
LLEND
LD19
LD21
LD23
LD10
LD12
LD14
LD3
LD5
LD7
LVM
C142
10uF/16V
C140
100nF
HEADCONN_40P
CON9
LD0
LD1
LD2
LD16
LD17
LD18
LLCD_PWREN
LD8
LD9
LCDVF0
LCDVF1
LCDVF2
VDD12V
C143
10uF/16V
HEADCONN_12P
CON8
TSYP
TSYM
TSXM
TSXP
BOXCONN_4P
C141
100nF
MITSHMI
M60-04-30-134P
SAMSUNG ELECTRONICS CO.,LTD
Title
SMDK2410X (S3C2410X DEMO BOARD)
Size
A3
Document Number
LCD
Rev
1.3
Date:
Monday, August 19, 2002
Sheet
8
of
12



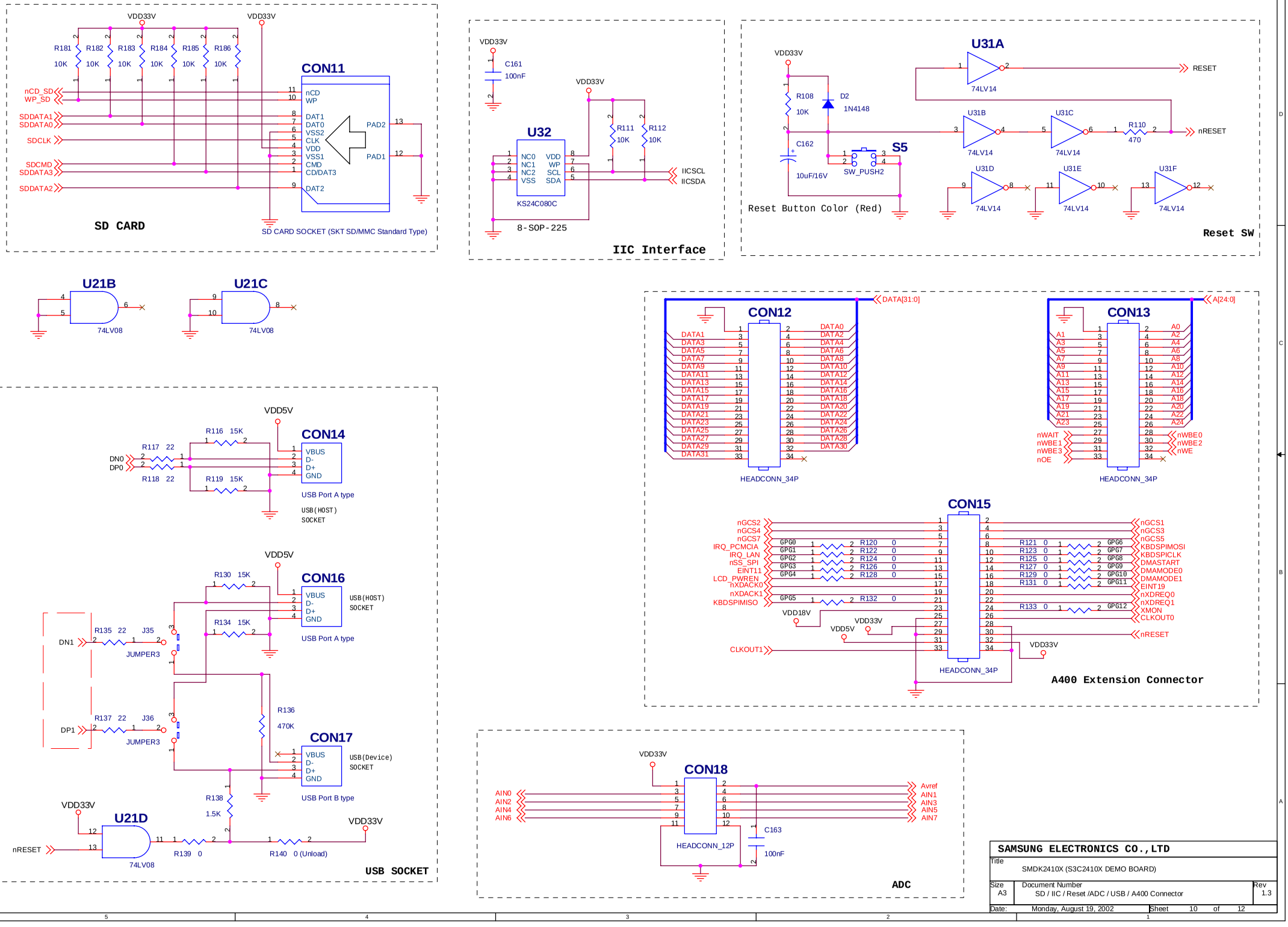


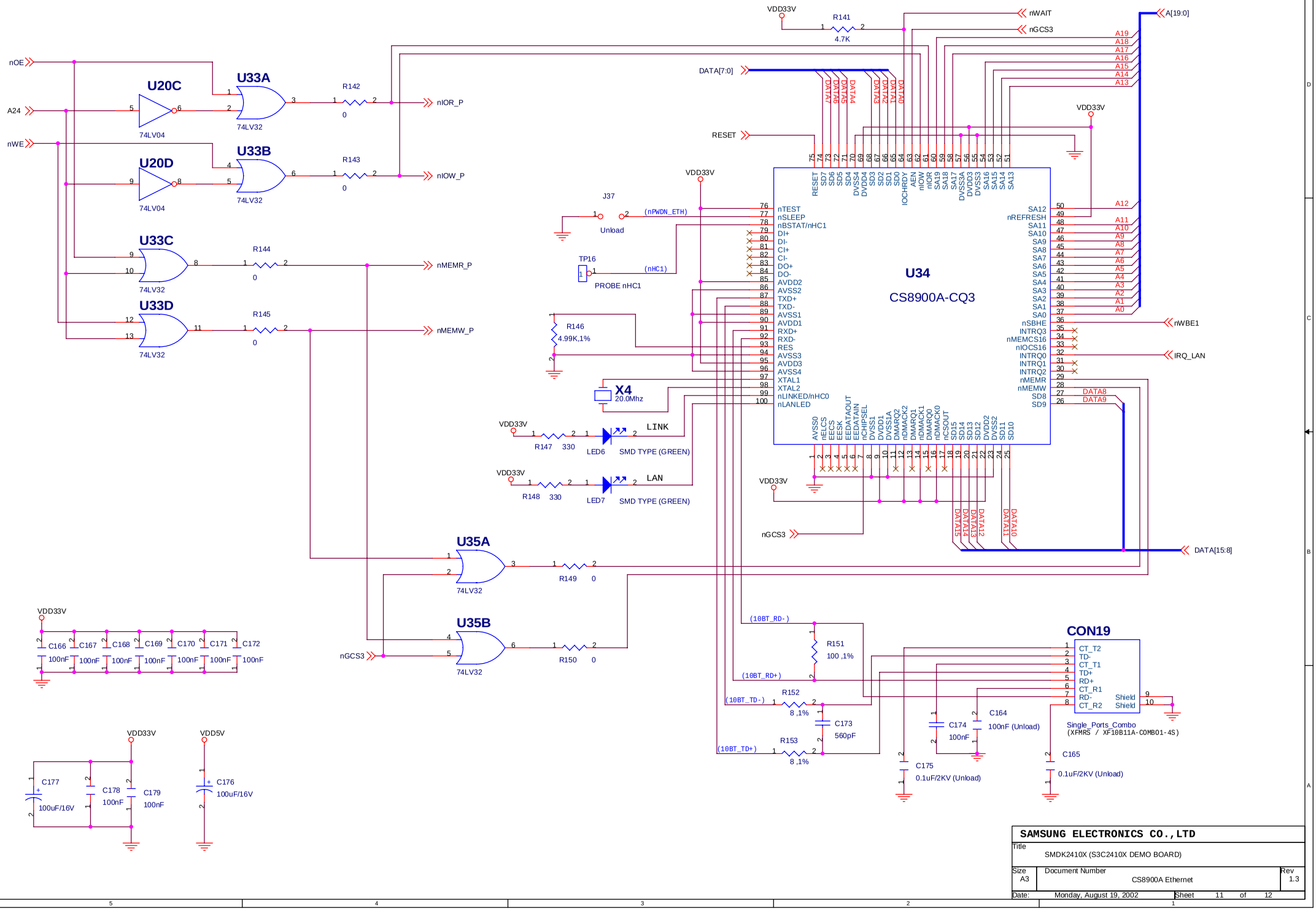




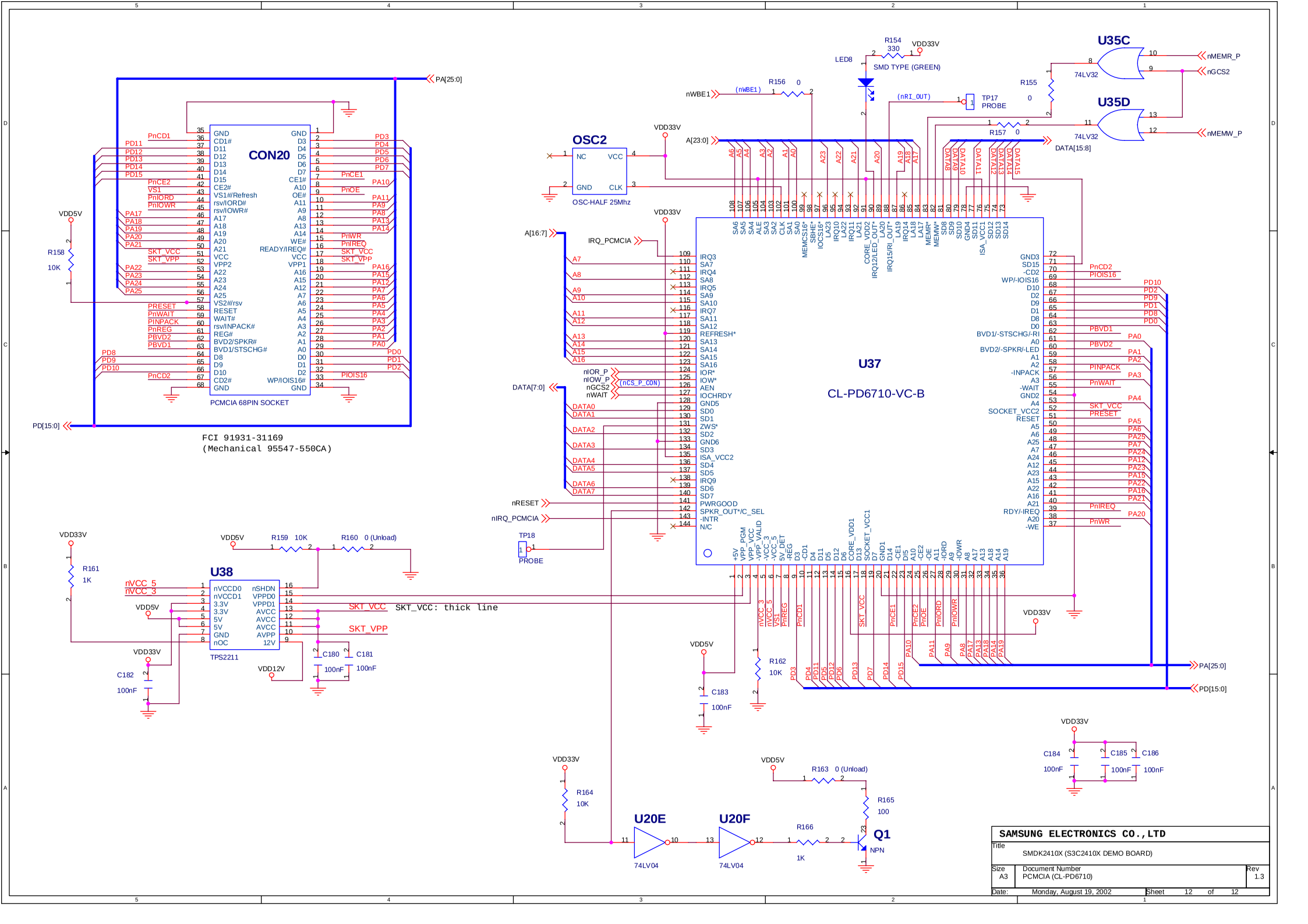


SAMSUNG ELECTRONICS CO.,LTD		
Title: SMDK2410X (S3C2410X DEMO BOARD)		
Size: A3	Document Number: POWER / SPI / JTAG/ LED / OM	Rev: 1.3
Date: Monday, August 19, 2002	Sheet 9 of 12	


SD CARD
SD CARD SOCKET (SKT SD/MMC Standard Type)
IIC Interface
Reset Button Color (Red)
Reset SW
74LV08
USB SOCKET
A400 Extension Connector
ADC
SAMSUNG ELECTRONICS CO.,LTD
SMDK2410X (S3C2410X DEMO BOARD)
SD / IIC / Reset /ADC / USB / A400 Connector
Monday, August 19, 2002



SAMSUNG ELECTRONICS CO.,LTD		
Title: SMDK2410X (S3C2410X DEMO BOARD)		
Size: A3	Document Number: CS8900A Ethernet	Rev: 1.3
Date: Monday, August 19, 2002	Sheet 11 of 12	


CON20
PCMCIA 68PIN SOCKET
FCI 91931-31169
(Mechanical 95547-550CA)
U37
CL-PD6710-VC-B
U38
TPS2211
OSC2
OSC-HALF 25Mhz
U35C
U35D
74LV32
U20E
U20F
74LV04
Q1
NPN
SMD TYPE (GREEN)
PROBE
SKT_VCC: thick line
SAMSUNG ELECTRONICS CO.,LTD
SMDK2410X (S3C2410X DEMO BOARD)
PCMCIA (CL-PD6710)
1.3
Monday, August 19, 2002
12 of 12

SMDK2410X(S3C2410X MAIN BOARD)

PCB Rev : 1.3 August 19, 2002

	Item	Description	Location	Qty
1	ATX POWER CONN_20P	RIGHT ANGLE(MOLEX)	CON10	1
2	Audio Codec	Philips UDA1341TS/N1 SSOP28	U16	1
3	CERAMIC CAP	0.33uF	C124	1
4	CERAMIC CAP	1nF	C187,188	2
5	CERAMIC CAP	100nF	C1~C8, C17~C26, C37~C90, C95~96, C126~C127, C129~C130 C132,C134~C141, C144, C148, C150 C154, C156, C158, C160~C161, C163 C166~C172, C174, C178~C186	113
6	CERAMIC CAP	100pF	C146, C152	2
7	CERAMIC CAP	150pF	C128	1
8	CERAMIC CAP	15pF	C113,C114	2
9	CERAMIC CAP	22pF	C91, C92, C119, C120	4
10	CERAMIC CAP	3.3nF	C125	1
11	CERAMIC CAP	330pF	C107,C108,C115~C118, C121,C122	8
12	CERAMIC CAP	560pF	C173	1
13	CONNECTOR	14P,KEYBOARD CONNECTOR	CON2	1
14	CONNECTOR	8P,KEYBOARD CONNECTOR	CON3	1
15	CONNECTOR	MITSHMI M60-04-30-134P, 4P	CON8	1
16	CONNECTOR DB9	RS232C CONNECTOR(RIGHT)	P1, P2	2
17	CONNECTOR HEAD	10P (5X2)	CON1	1
18	CONNECTOR HEAD	12P (6X2)	CON9, CON18	2
19	CONNECTOR HEAD	20P (10X2)	CON5	1
20	CONNECTOR HEAD	34P	CON12,13,15	3
21	CONNECTOR HEAD	40P (20X2)	CON7	1
22	CONNECTOR HEAD	5P	CON4	1
23	CONNECTOR HEAD BOX	20P(10X2),MULTI_ICE	CN2	1
24	DIODE	1N4148	D2	1
25	DIODE	MBRS130L	D1	1
26	FET	FAIRCHILD FDC6321 SOP	U26,28	2
27	IC	ALTERA EPM7032AETC44-7 44TQFP	U4	1
28	IC	TI 74LV08 14SOP	U21	1
29	IC Bus Transceiver	IDT74LVCH162245APA 48TSOP	U2,3,5,6,7	5
30	IC Bus Transceiver	TI SN74LVC4245ADBR 24SSOP	U22,24,25,27	4
31	IC DC DC Converter	MAXIM MAX629	U23	1
32	IC EEPROM	SEC S524C80D80 (Old KS24C080C), 8SOP (SEC S524C80D81 (Old KS24C081C)	U32	1
33	IC Hex Inverter	TI 74LV04 14SOP	U20	1
34	IC INVERTER	TI 74LV14 , 14SOP	U31	1
35	IC ISA ETHERNET CONTROLL	CRYSTAL CS8900A-CQ3 QFP	U34	1
36	IC OR-GATE	TI 74LV32, 14SOP	U33,U35	2
37	IC PCMCIA CONTROLLER	CIRRUS CL-PD6710-VC-B QFP	U37	1
38	IC POWER INTERFACE	TI TPS2211 SOP	U38	1
39	IC SOCKET	TI CTP048-111AB-2 48TSOP	U10	1
40	INDUCTOR	47uH	L1	1
41	INDUCTOR	SMD BEAD	B1,B2	2
42	IrDA	ROHM RPM851A	U19	1
43	JACK	AUDIO JACK	CN1	1
44	JUMPER	2 P	J1,J3, J10,J11,J16,J17,J21~J34	20
45	JUMPER	3 P	J2,4~J7,J9,J12~J15,J18~J20,J35,J36	15
46	JUMPER SHUNT	2 P		30
47	Keyboard Encoder	SAMTECH UR5HCSPI-SA 44QFP	U15	1
48	LED	SMD TYPE (GREEN)	LED1,2,3,4,6,7,8	7
49	LED	SMD TYPE (RED)	LED5	1
50	LINEAR REGULATORS	NSC LP3965EMP-ADJ, TO263	U29,30	2

	Item	Description	Location	Qty
51	MCU	S3C2410X	U1	1
52	MCU Socket	Enplas OTB-280(289)-0.8-12	U1	1
53	MICROPHONE	SKP446(1.5V, 60dB)	MIC1	1
54	NAND Flash	SAMSUNG 64MB Smart Media Card	CON21	1
55	NOR Flash	AM29LV800BB-90EC 1MB 48TSOP	U10	1
56	OSC-HALF	25Mhz	OSC2	1
57	PCB	8 Layer, 2.0t		1
58	PCMCIA SOCKET	FCI, 68P	CON20	1
59	Poly Switch	miniSMDC150(FUSE)	F1	1
60	POWER S/W	TOGGLE SW	SW1	1
61	PROBE		TP1,3,9~18	12
62	RESISTOR	0	R4,5,7,8,12,13,22,23,24,34,37,44,45,52,53 R57,58,59,60,61,63,69,75,99,102, R120~129,131~133 R139,142~145,149,150,155~157,170	49
63	RESISTOR	10	R172, R180	2
64	RESISTOR	22	R6,117,118,135,137,173~179	12
65	RESISTOR	100	R46,47,48,165	4
66	RESISTOR	330	R79,82,101,105,147,148,154	7
67	RESISTOR	470	R98, 110	2
68	RESISTOR	1.5K	R138	1
69	RESISTOR	10K	R15,16,17,31,32,50,51,64,65,78,80,83 R84,92,93,94,95,108,111,112,158,159,162 R164,181 ~ 186	30
70	RESISTOR	150K	R35,38,41	3
71	RESISTOR	15K	R39,40,43,116,119,130,134	7
72	RESISTOR	1K	R85,86,91,96,161,166	6
73	RESISTOR	1M	R42	1
74	RESISTOR	4.7 K	R1,2,3,18,19,20,21,28,29,30,33,54,55, R56,68,70,72,73,74,76,77,81,87,88,89 R90,97,100,103,104,141,167,168,171	34
75	RESISTOR	4.99K, 1%	R146	1
76	RESISTOR	470K	R136	1
77	RESISTOR	6.8K	R49	1
78	RESISTOR	68K	R71	1
79	RESISTOR ARRAY	22 ohm	RA1,RA2	2
80	RESISTOR DIP	5.6	R66	1
81	RESISTOR DIP	47	R67	1
82	RESISTOR DIP	100, 1%	R151	1
83	RESISTOR DIP	8, 1%	R152,153	2
84	SD CARD SOCKET	SKT SD/MMC Standard Type	CON11	1
85	SDRAM	K4S561632C-TC/L1H (32M) 56TSOP	U8,9	2
86	SINGLE PORT COMBO	XF10B11A-COMBO1-2(4)S	CON19	1
87	SOCKET SMART MEDIA	YAMAICHI CN015R-3123-0	CON21	1
88	SWITCH BUTTON	TACT SW (S5:Red,S1~4:Black)	S1,2,3,4,5	5
89	TANTAL	0.1uF/16V	C101~C106, C109, C110	8
90	TANTAL	100uF/16V	C93,C149,C155,C157,C159,C176,C177	7
91	TANTAL	10uF/16V	C9~C16, C27~C36, C133, C142~C143 C145, C147, C152, C153, C162	26
92	TANTAL	22uF/36V (or 10uF/36V)	C131	1
93	TANTAL	4.7uF/16V	C123	1
94	TANTAL	47uF/16V	C94,C97,C98,C99,C100	5
95	Transceiver RS232	MAX3232C(SIPEX SP232) 16TSSOP	U17,18	2
96	TRANSISTOR	KSC2895Y, SOT23	Q1	1

	Item	Description	Location	Qty
97	USB PORT	B TYPE	CON17	1
98	USB PORT	A TYPE	CON14,16	2
99	VR (MULTI TURN)	10K	VR1	1
100	X-TAL	12Mhz	X2	1
101	X-TAL	20MHz	X4	1
102	X-TAL	2MHz	X1	1
103	X-TAL	32.768KHz	X3	1

518

* Unload : R7,9,10,11,14,25,56,57,36,62,140,160,163
C111,C112,C164,C165,C175
U12,U13,U14, J37, OSC1
CON6(MOLEX, 52207-1690)

Keyboard
Power Supply
Serial Cable
Power Cord
USB Cable

Revision History	Old	New
Ver : 1.3 (20020703)	J2 2Pin	3Pin
	R106,R107,R109,R113~R115 0 ohm	Deleted
	-	R181 ~ R186 10Kohm

SMDK2410X(S3C2410X LCD BOARD ASS'Y)

PCB Rev : 1.2 June 26, 2002

	Item	Description	Location	Qty
1	AMPLIFIERS	AD8541AR	U503	1
2	AMPLIFIERS	AD8542	U502	1
3	CAPACITOR DIP	220uF/16V	C530	1
4	CERAMIC CAP	0.1uF	C503,504,509,510,511,516 ,522,525,529	9
5	CERAMIC CAP	100pF	C527	1
6	CERAMIC CAP	10pF	C501	1
7	CERAMIC CAP	10uF	C507,506	2
8	CERAMIC CAP	1uF	C502,505,508,512,513	5
9	CERAMIC CAP	22pF	C524	1
10	CONNECTOR DIP 12P	HEADCONN_12P, MALE	CON504	1
11	CONNECTOR DIP 3P	HEADCONN_3P, MALE	JP501	1
12	CONNECTOR DIP 40P	HEADCONN_40P, MALE	CON503	1
13	CONNECTOR DIP 4P	HEAD_1WALL_4P,RIGHT ANGLE	CON506	1
14	CONNECTOR SMD 4P	HIROSE FH12-10(4)SA-1SH	CON505	1
15	CONNECTOR SMD 4P	JST SM04B-SRSS-TB	CON502	1
16	CONNECTOR SMD 50P	HIROSE FH12K-50S-0.5SH	CON501	1
17	DIODE	BAT750	D501	1
18	DIODE	MMBD4448HTW	D502	1
19	DIODE	MMDT2227	Q501	1
20	GENERATORS	MAXIM MAX1605	U501	1
21	HARNESS ASS'Y	Inverter to LCD B'd		1
22	INDUCTOR	SUMIDA CMD4D08	L501	1
23	INVERTER BOARD	SEM SIP-350		1
24	LCD	SAMSUNG LTS350Q1-PD1		1
25	PCB	4 Layer, 2.0t		1
26	REGULATOR	NATIONAL LP365EMP-ADJ	U504	1
27	RESISTOR	0	R540,541,542	3
28	RESISTOR	100	R538,521	2
29	RESISTOR	220	R529	1
30	RESISTOR	330	R549	1
31	RESISTOR	360	R532	1
32	RESISTOR	390	R533,R528	2
33	RESISTOR	510	R531	1
34	RESISTOR	680	R530,R512	2
35	RESISTOR	1.2K	R526,536	2
36	RESISTOR	100K	R501,523,534,535	4
37	RESISTOR	10K	R507,508,519,520,548,551	6
38	RESISTOR	12K	R502,510,518	3
39	RESISTOR	15K	R503	1
40	RESISTOR	18K	R524	1
41	RESISTOR	1M ohm	R522	1

	Item	Description	Location	Qty
42	RESISTOR	2.2K	R537	1
43	RESISTOR	27K	R511	1
44	RESISTOR	30K	R513,514,516	3
45	RESISTOR	36K	R505,515	2
46	RESISTOR	39K	R517	1
47	RESISTOR	51K	R506	1
48	RESISTOR	560K	R525	1
49	RESISTOR	56K	R509	1
50	RESISTOR	6.8K	R550	1
51	RESISTOR	68K	R504	1
52	SHUNT	2P	JP501	1
53	TANTAL	100uF/16V	C526,528	2

* Unload : R539, CON507 (MOLEX 52974-0207)

SMDK2410X(S3C2410X LCD Interface Board)

PCB Rev : 1.0 June 26, 2002

	Item	Description	Location	Qty
1	CONNECTOR DIP 40P	HEADCONN_40P, FEMALE	CON3, CON1	2
2	CONNECTOR DIP 12P	HEADCONN_12P, FEMALE	CON2, CON4	2
3	PCB	2 Layer, 2.0t		1