SEC JTAG FLASH(SJF) USER'S GUIDE

DATE:NOV.15.2002

SJF2410 can program SMDK2410 flash memory (K9S1208,Intel E28F128,AMD29LV800BB) through JTAG port and read/write data from/to a specified address.

SJF2410 VERSION HISTORY

Version	Description
0.1	K9S1208 programming is supported.
0.2	External peripherals (PD6710,CS8900A and etc) read/write is supported .
0.3	Intel E28F128 StrataFlash programming is supported.
0.4	- AMD 29LV800BB programming is supported. - E28F128 StrataFlash programming speed is more fast. - E28F128 ClearLockBit function is added before sector erase.

INSTALLING GIVEIO.SYS

In windows NT/2000/XP, any application can't access the I/O such as the parallel port. So, GIVEIO.SYS enables SJF.exe to access the parallel port without any memory fault. In windows 95/98, GIVEIO.SYS isn't needed.

For Windows* 2000, use the following procedure:

1) Login as administrator
2) Copy the giveio.sys file to %systemroot%\system32\drivers.
3) Choose Control Panel, and choose Add/Remove Hardware.
4) Select 'Add/Troubleshoot a device'
5) Select 'Add a new device' and choose Next, and select 'No, I want to select the hardware from a list'
6) Select Other devices and choose 'Have Disk...'.
7) Choose 'Browse...', locate the folder where giveio.inf file.
8) Complete the remained process.

For Windows* NT, use the following procedure:

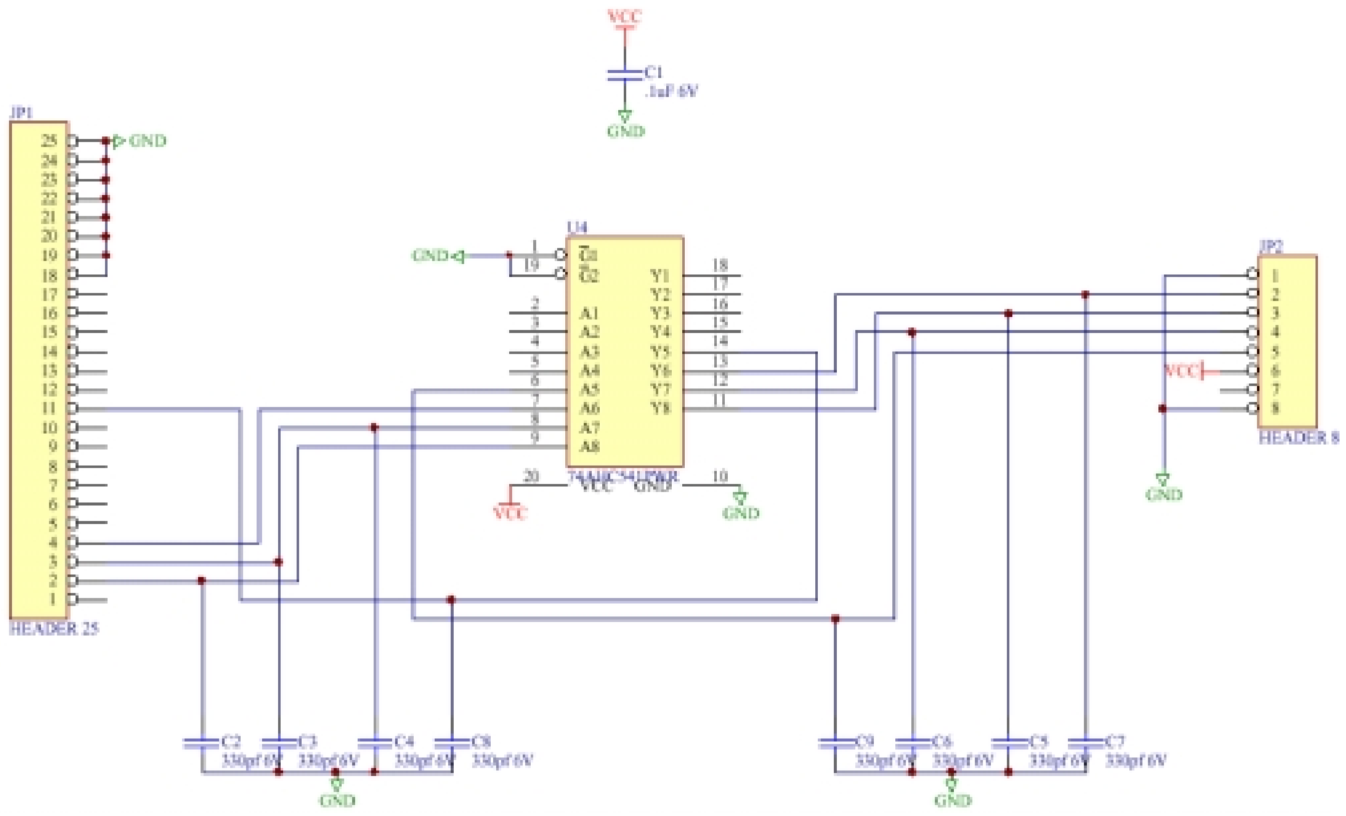
1) Login as administrator.
2) Open a DOS command window.
3) Copy giveio.sys to %systemroot%\system32\drivers.
4) Install the driver using the instdrv utility specifying the driver name and the FULL PATH NAME to the giveio.sys file.
   ```
   instdrv giveio c:\winnt\system32\drivers\giveio.sys
   ```
5) To enable the driver to start automatically each time you boot, use the following procedure:
   - Choose Settings and choose Control Panel.
   - Choose Devices, select giveio from the list, and choose Startup.
   - Select Startup Type Automatic from the Device menu.

INSTALLING JTAG DONGLE

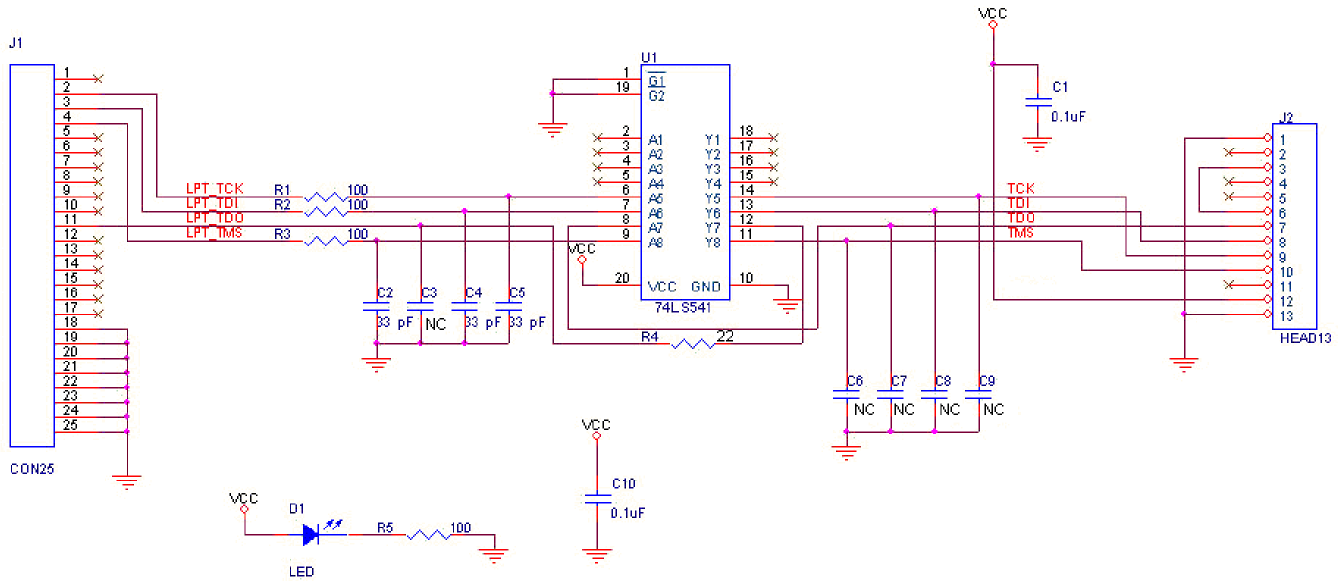
The JTAG dongle is JTAG programming cable, which is connected to the PC parallel port.

Because we can't get 74AHC541, 74LS541 is used instead. So, some circuit is modified from the original dongle circuit from http://www.lart.tudelft.nl/projects/jtag/. (This dongle can be also used for SA-1110.)

It's recommended to refer to the original JTAG dongle circuit. If you can't get 74AHC541, refer to our schematic circuit. If you are using our schematic, the cable length should not be longer than 1m.



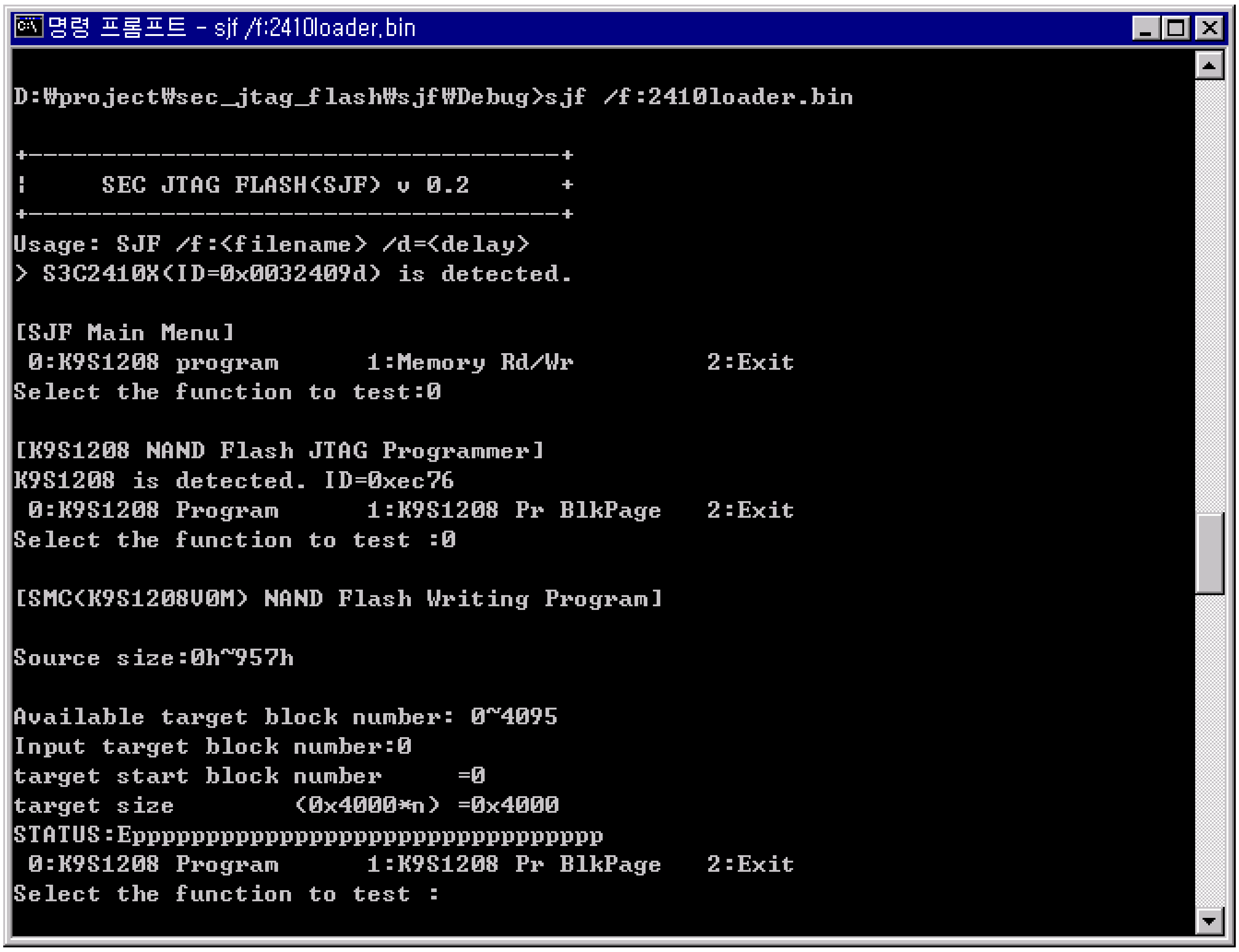
Holly Gates' Schematics for the JTAG Dongle (http://www.lart.tudelft.nl/projects/jtag/)



Modified Schematic for the JTAG Dongle Circuit Using 74LS541

TO PROGRAM BOOT LOADER CODE ON K9S1208 NAND FLASH

1) Prepare your own boot loader image.(For example, 2410loader.bin is used here)
2) Run SJF.exe in the DOS command window as following example.
   `SJF2410 /f:2410loader.bin`
3) Type as follows

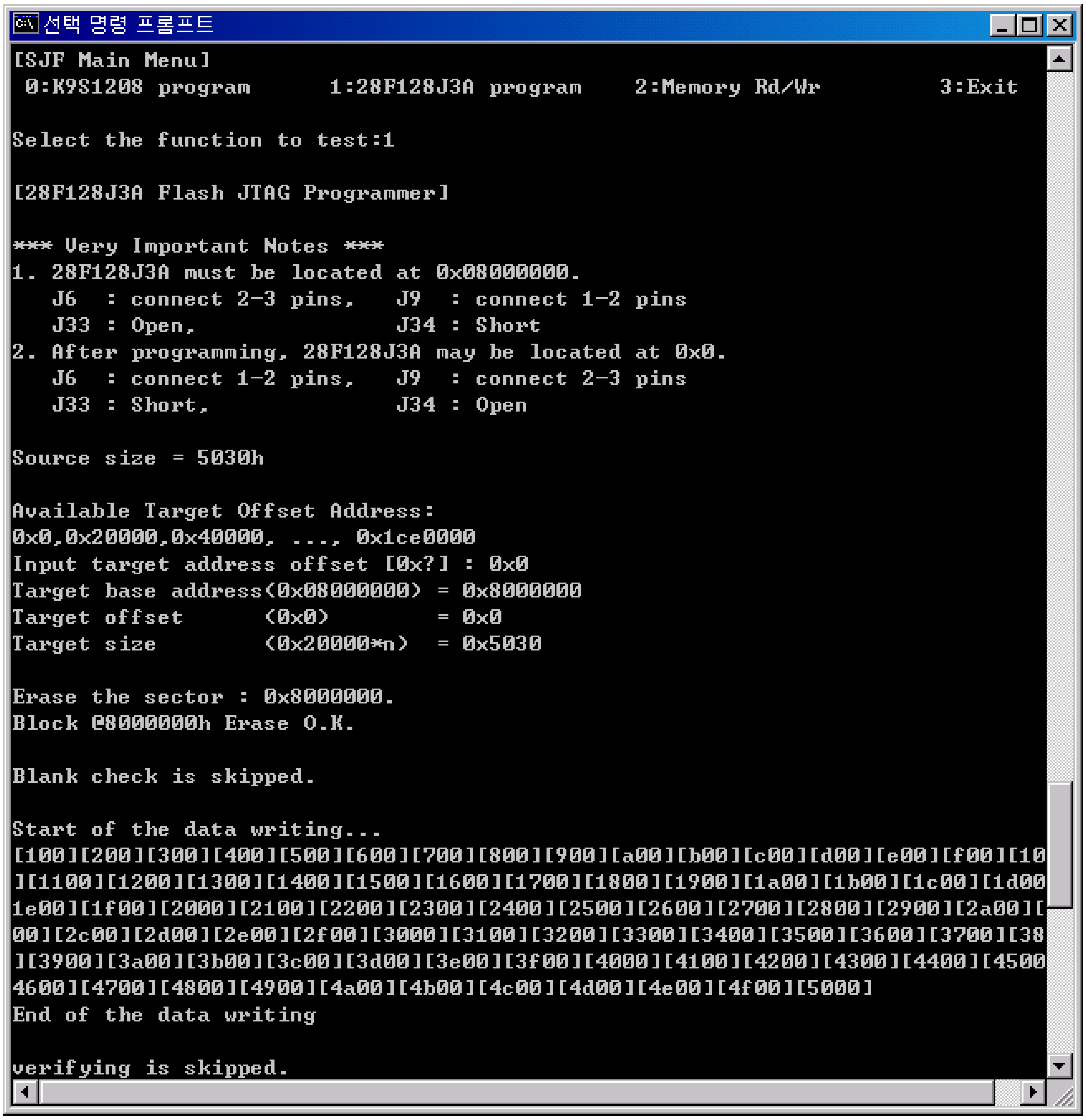


```
D:\project\sec_jtag_flash\sjf\Debug>sjf /f:2410loader.bin

+------------------------------------+
|     SEC JTAG FLASH(SJF) v 0.2      +
+------------------------------------+
Usage: SJF /f:<filename> /d=<delay>
> S3C2410X(ID=0x0032409d) is detected.

[SJF Main Menu]
 0:K9S1208 program      1:Memory Rd/Wr         2:Exit
Select the function to test:0

[K9S1208 NAND Flash JTAG Programmer]
K9S1208 is detected. ID=0xec76
 0:K9S1208 Program      1:K9S1208 Pr BlkPage   2:Exit
Select the function to test :0

[SMC(K9S1208V0M) NAND Flash Writing Program]

Source size:0h~957h

Available target block number: 0~4095
Input target block number:0
target start block number     =0
target size        (0x4000*n) =0x4000
STATUS:Epppppppppppppppppppppppppppppppp
 0:K9S1208 Program      1:K9S1208 Pr BlkPage   2:Exit
Select the function to test :
```

(The above screen was captured from SJF2410 v0.2)

TO PROGRAM BOOT CODE ON 28F128 x 2 STRATA FLASH

1) Prepare your own boot loader image.(For example, 2410mon.bin is used here)
2) Run SJF.exe in the DOS command window as following example.
   `SJF2410 /f:2410mon.bin`
3) Type as follows

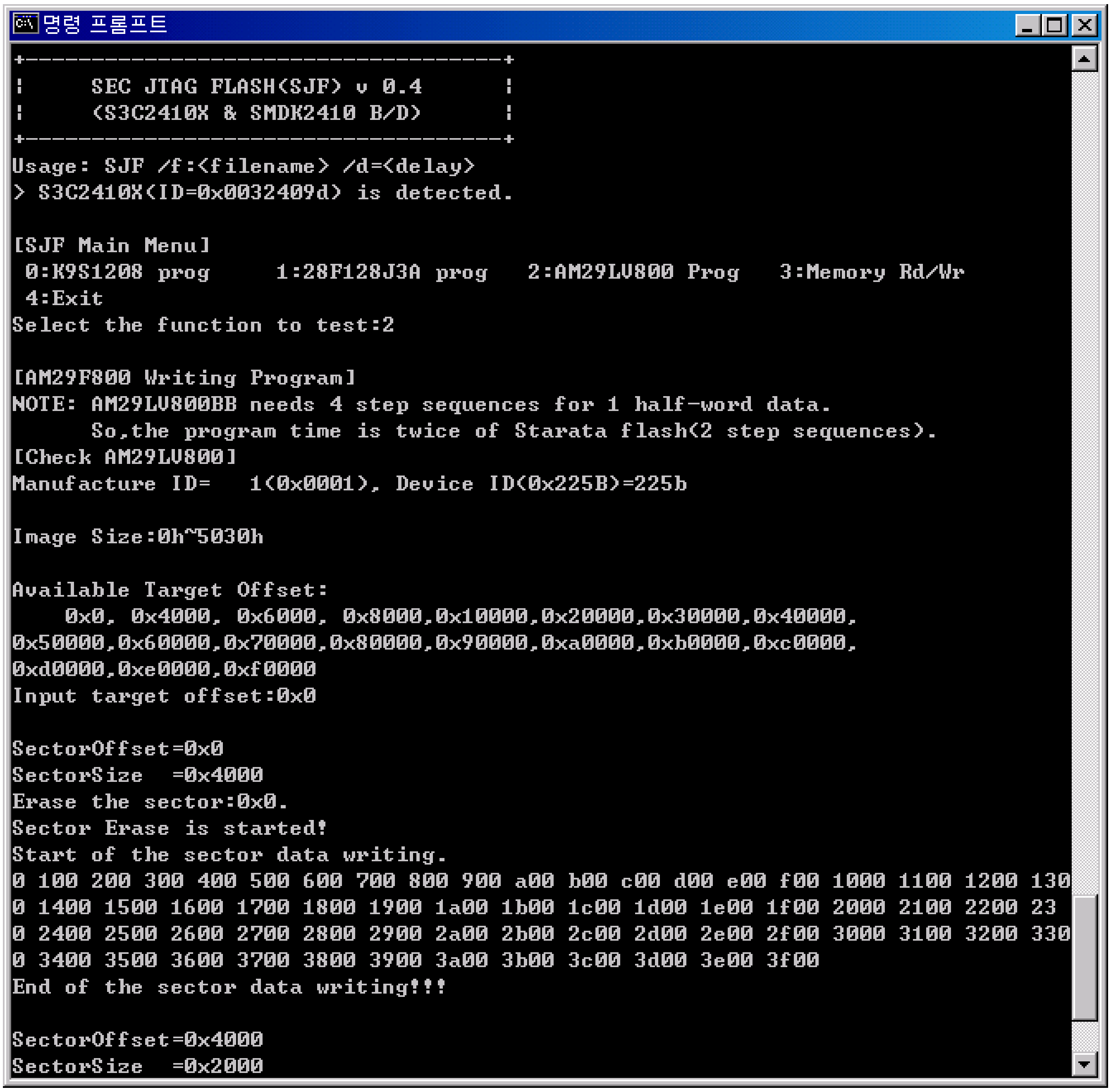


```
[SJF Main Menu]
 0:K9S1208 program      1:28F128J3A program    2:Memory Rd/Wr         3:Exit

Select the function to test:1

[28F128J3A Flash JTAG Programmer]

*** Very Important Notes ***
1. 28F128J3A must be located at 0x08000000.
   J6  : connect 2-3 pins,    J9  : connect 1-2 pins
   J33 : Open,                J34 : Short
2. After programming, 28F128J3A may be located at 0x0.
   J6  : connect 1-2 pins,    J9  : connect 2-3 pins
   J33 : Short,               J34 : Open

Source size = 5030h

Available Target Offset Address:
0x0,0x20000,0x40000, ...., 0x1ce0000
Input target address offset [0x?] : 0x0
Target base address(0x08000000) = 0x8000000
Target offset       (0x0)        = 0x0
Target size         (0x20000*n)  = 0x5030

Erase the sector : 0x8000000.
Block @8000000h Erase O.K.

Blank check is skipped.

Start of the data writing...
[100][200][300][400][500][600][700][800][900][a00][b00][c00][d00][e00][f00][10
][1100][1200][1300][1400][1500][1600][1700][1800][1900][1a00][1b00][1c00][1d00
1e00][1f00][2000][2100][2200][2300][2400][2500][2600][2700][2800][2900][2a00][
00][2c00][2d00][2e00][2f00][3000][3100][3200][3300][3400][3500][3600][3700][38
][3900][3a00][3b00][3c00][3d00][3e00][3f00][4000][4100][4200][4300][4400][4500
4600][4700][4800][4900][4a00][4b00][4c00][4d00][4e00][4f00][5000]
End of the data writing

verifying is skipped.
```

(The above screen was captured from SJF2410 v0.3)

TO PROGRAM BOOT CODE ON AM29LV800BB FLASH

1) Prepare your own boot loader image.(For example, 2410mon.bin is used here)
2) Run SJF.exe in the DOS command window as following example.
   `SJF2410 /f:2410mon.bin`
3) Type as follows

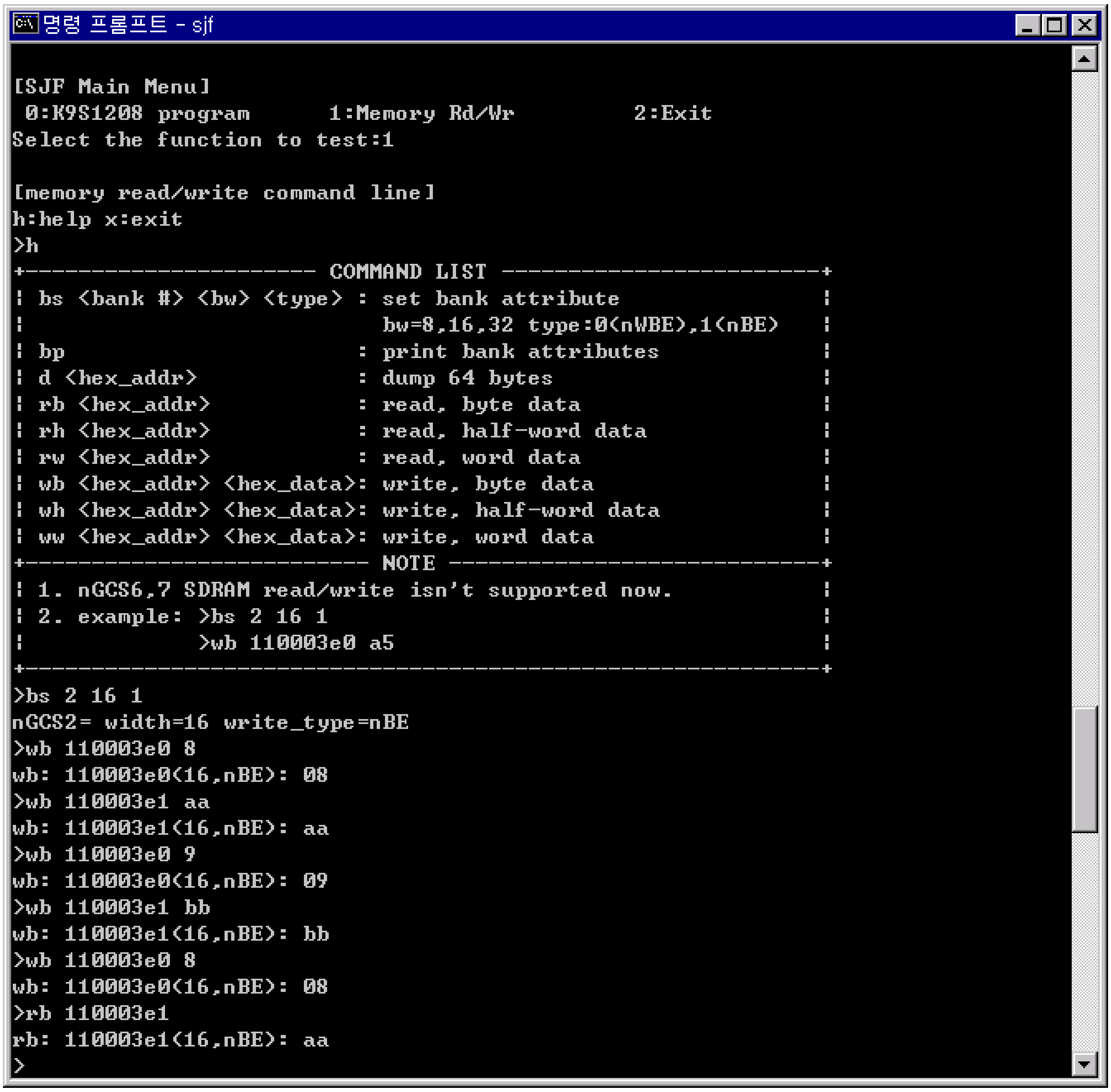


```
+--------------------------------------+
|     SEC JTAG FLASH(SJF) v 0.4        |
|     (S3C2410X & SMDK2410 B/D)        |
+--------------------------------------+
Usage: SJF /f:<filename> /d=<delay>
> S3C2410X(ID=0x0032409d) is detected.

[SJF Main Menu]
 0:K9S1208 prog     1:28F128J3A prog    2:AM29LV800 Prog    3:Memory Rd/Wr
 4:Exit
Select the function to test:2

[AM29F800 Writing Program]
NOTE: AM29LV800BB needs 4 step sequences for 1 half-word data.
      So,the program time is twice of Starata flash(2 step sequences).
[Check AM29LV800]
Manufacture ID=   1(0x0001), Device ID(0x225B)=225b

Image Size:0h~5030h

Available Target Offset:
    0x0, 0x4000, 0x6000, 0x8000,0x10000,0x20000,0x30000,0x40000,
0x50000,0x60000,0x70000,0x80000,0x90000,0xa0000,0xb0000,0xc0000,
0xd0000,0xe0000,0xf0000
Input target offset:0x0

SectorOffset=0x0
SectorSize  =0x4000
Erase the sector:0x0.
Sector Erase is started!
Start of the sector data writing.
0 100 200 300 400 500 600 700 800 900 a00 b00 c00 d00 e00 f00 1000 1100 1200 130
0 1400 1500 1600 1700 1800 1900 1a00 1b00 1c00 1d00 1e00 1f00 2000 2100 2200 23
0 2400 2500 2600 2700 2800 2900 2a00 2b00 2c00 2d00 2e00 2f00 3000 3100 3200 330
0 3400 3500 3600 3700 3800 3900 3a00 3b00 3c00 3d00 3e00 3f00
End of the sector data writing!!!

SectorOffset=0x4000
SectorSize  =0x2000
```

(The above screen was captured from SJF2410 v0.4)

TO READ/WRITE A SPECIFIED ADDRESS OF THE EXTERNAL MEMORY BUS

1) SJF.exe in the DOS command window as following example.
   `SJF`
   or `SJF2410`
2) Refer to the following example is to read/write the PD6710 register in SMDK2410 board.



```
[SJF Main Menu]
 0:K9S1208 program      1:Memory Rd/Wr         2:Exit
Select the function to test:1

[memory read/write command line]
h:help x:exit
>h
+----------------------- COMMAND LIST -------------------------+
| bs <bank #> <bw> <type> : set bank attribute                  |
|                            bw=8,16,32 type:0(nWBE),1(nBE)     |
| bp                      : print bank attributes               |
| d <hex_addr>            : dump 64 bytes                       |
| rb <hex_addr>           : read, byte data                     |
| rh <hex_addr>           : read, half-word data                |
| rw <hex_addr>           : read, word data                     |
| wb <hex_addr> <hex_data>: write, byte data                    |
| wh <hex_addr> <hex_data>: write, half-word data               |
| ww <hex_addr> <hex_data>: write, word data                    |
+--------------------------- NOTE -----------------------------+
| 1. nGCS6,7 SDRAM read/write isn't supported now.              |
| 2. example: >bs 2 16 1                                        |
|             >wb 110003e0 a5                                   |
+--------------------------------------------------------------+
>bs 2 16 1
nGCS2= width=16 write_type=nBE
>wb 110003e0 8
wb: 110003e0(16,nBE): 08
>wb 110003e1 aa
wb: 110003e1(16,nBE): aa
>wb 110003e0 9
wb: 110003e0(16,nBE): 09
>wb 110003e1 bb
wb: 110003e1(16,nBE): bb
>wb 110003e0 8
wb: 110003e0(16,nBE): 08
>rb 110003e1
rb: 110003e1(16,nBE): aa
>
```

(The above screen was captured from SJF2410 v0.2)