

S

ECJTAG FLASH(SJF) USER'S GUIDE

(PRELIMINARY)

DATE:OCTOBER 31, 2005

SJF2413 can program SMDK2413 flash memory (K9S1208, AMD29LV800BB) through JTAG port and read/write data from/to a specified address.

SJF2413 VERSION HISTORY

Version	Description
0.1	- K9S1208 programming is supported. - AMD 29LV800BB programming is supported.

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:

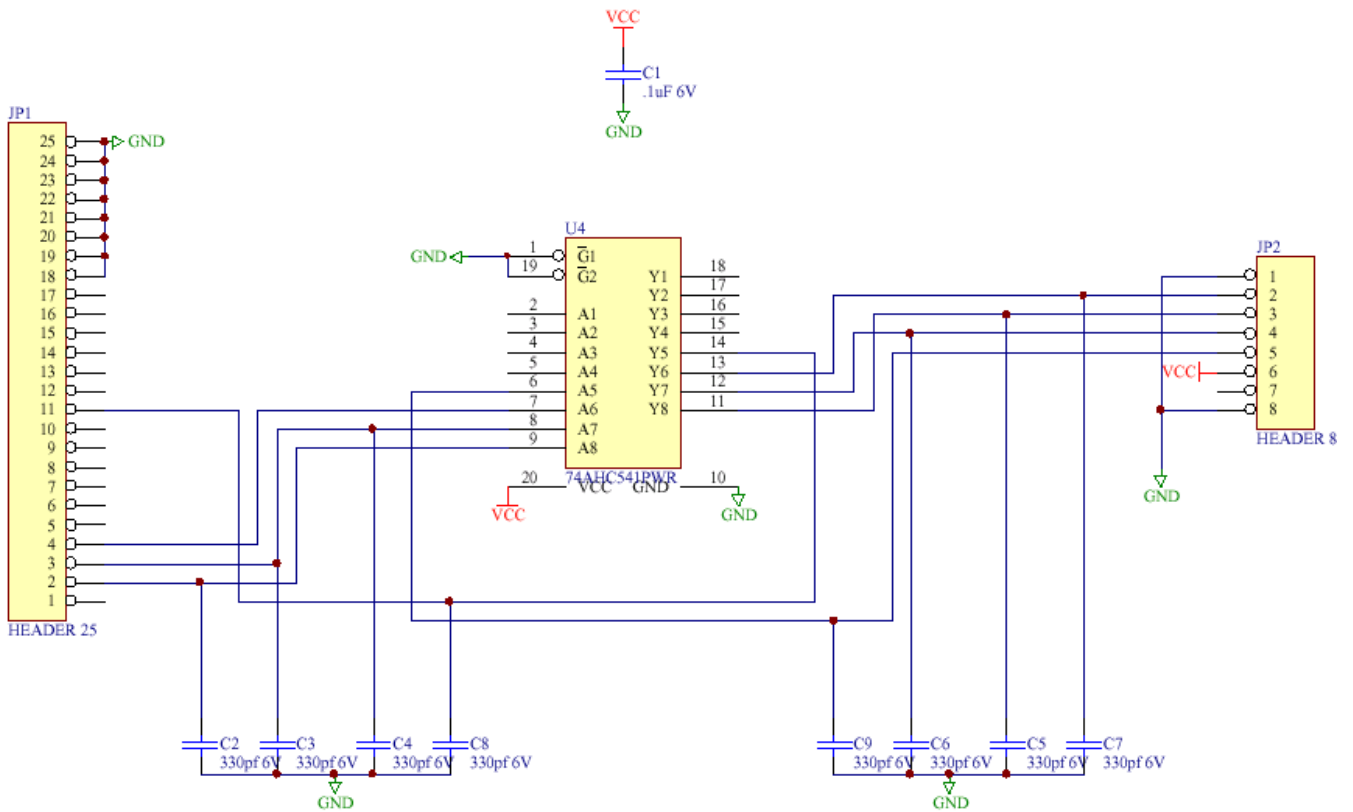
- 1) Login as administrator.
- 2) Open a DOS command window.
- 3) Copy giveio.sys to %systemroot%\system32\drivers.
- 4) Install the driver using the instdrv utility specifying the driver name and the FULL PATH NAME to the giveio.sys file.
`instdrv giveio c:\winnt\system32\drivers\giveio.sys`
- 5) To 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

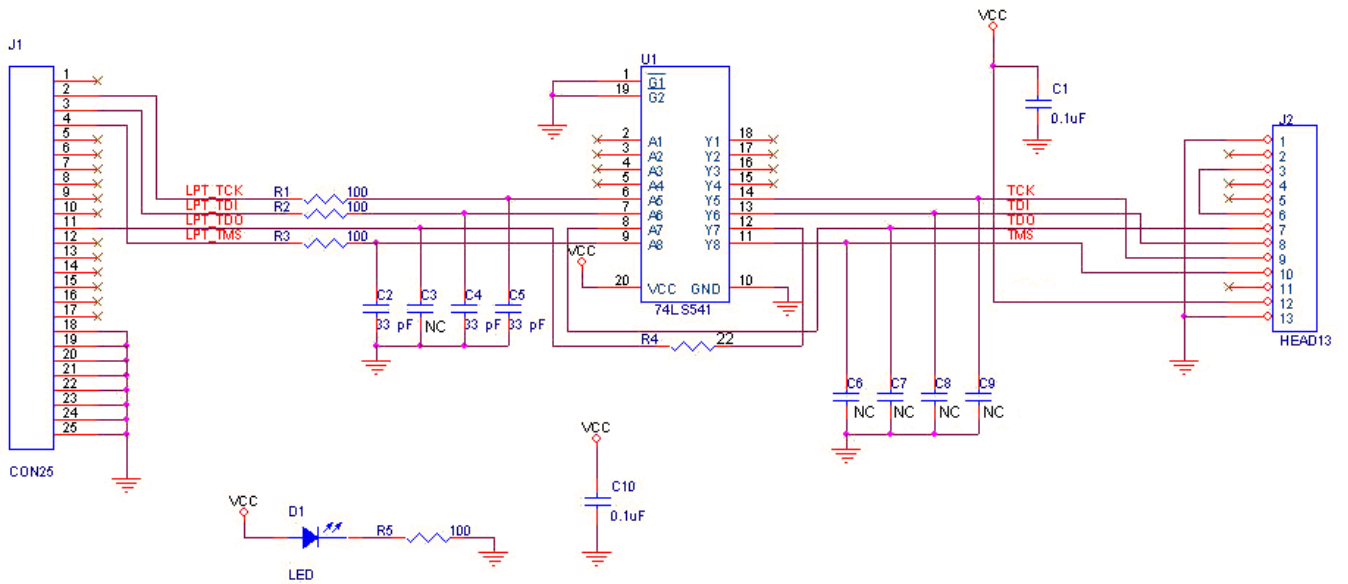
The JTAG dongle is JTAG programming cable, which is connected to the PC parallel port.

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.



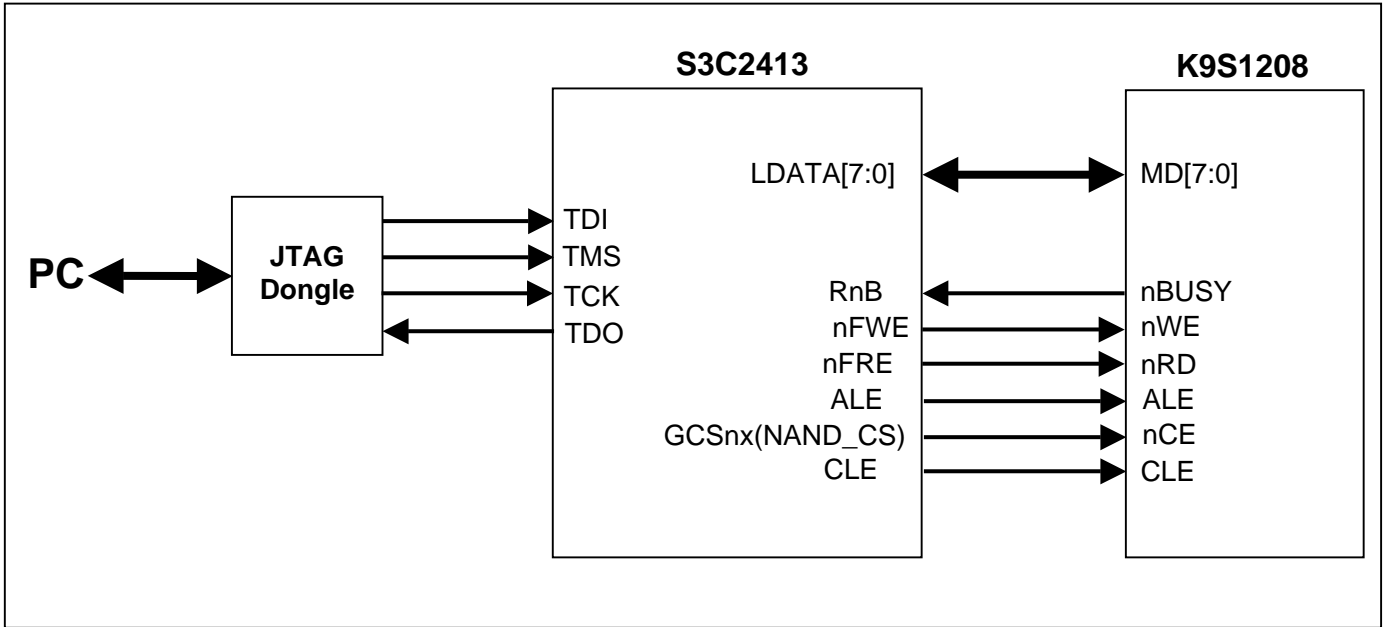
Holly Gates' Schematics for the JTAG Dongle (<http://www.lart.tudelft.nl/projects/jtag/>)



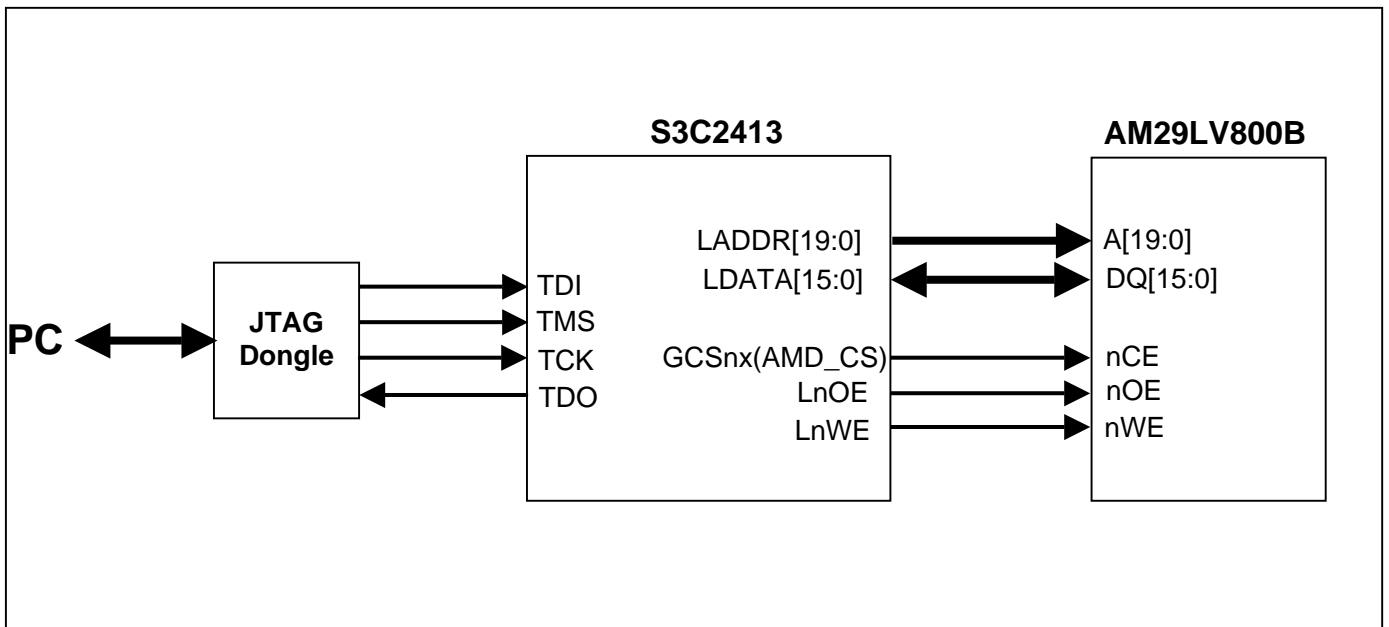
Modified Schematic for the JTAG Dongle Circuit Using 74LS541

HARDWARE INTERFACE FOR FLASH PROGRAMMING USING SJF2413

When you program flash memory using SJF2413, you should consider following hardware interface.

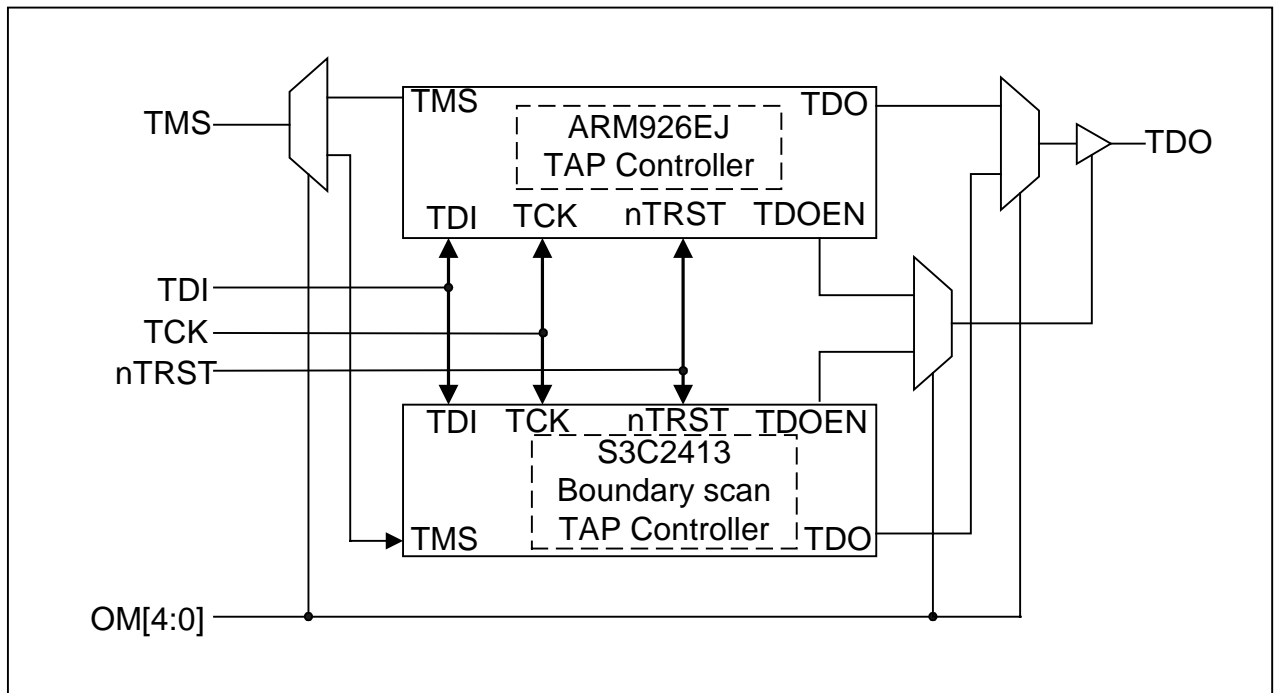


Hardware Interface for K9S1208 NAND Flash Programming



Hardware Interface for AM29LV800BB NOR Flash Programming

INFORMATION OF S3C2413 SCAN CHAIN



TO CHECK S3C2413 CHIP ID

- 1) IF [OM4:0] is not "10111".
- 2) Run 2413dongle.exe in the DOS command window as following example.
2413dongle

Type as follows

```

명령 프롬프트
E:\work\product\ARMW2413\program\sjf\2413Dongle\2413dongle\Debug>2413dongle /f:2413loader8bit.bin
+-----+
| SEC JTAG FLASH(SJF) v 0.0 |
| (S3C2413 & SMDK2413 B/D) |
+-----+
Usage: SJF /f:<filename> /d=<delay>
Image size:c38
> ARM926EJ(ID=0x0792609d) is detected.
> You cannot program memory read/write
> If you want to program memory read/write, OM[4:0] should be set '10111'
E:\work\product\ARMW2413\program\sjf\2413Dongle\2413dongle\Debug>_
    
```


TO PROGRAM BOOT CODE ON AM29LV800BB FLASH(OM[4:0] = "10111").

- 1) Prepare your own boot loader image.(For example, 2413_uart_mon.bin is used here)
- 2) Run 2413dongle.exe in the DOS command window as following example.
2413dongle /f:2413_uart_mon.bin
- 3) Type as follows

```

C:\ 명령 프롬프트
E:\work\product\WARMW2413\program\sjf\W2413Dongle\W2413dongle\Debug>2413dongle /f:2413_u
art_mon.bin

+-----+
|      SEC JTAG FLASH<SJF> v 0.0      |
|      <S3C2413 & SMDK2413 B/D>      |
+-----+
Usage: SJF /f:<filename> /d=<delay>
Image size:3950
> ARM926EJ<ID=0x0792609d> is detected.
> You cannot program memory read/write
> If you want to program memory read/write, OM[4:0] should be set '10111'

E:\work\product\WARMW2413\program\sjf\W2413Dongle\W2413dongle\Debug>2413dongle /f:2413_u
art_mon.bin

+-----+
|      SEC JTAG FLASH<SJF> v 0.0      |
|      <S3C2413 & SMDK2413 B/D>      |
+-----+
Usage: SJF /f:<filename> /d=<delay>
Image size:3950
> S3C2413<ID=0x00c8509d> is detected.
> You can program memory read/wirte

[ SJF Main Menu ]
0:K9S1208 prog      1:AM29L800 Prog      2:Exit
Select the function to test:1

[ AM29F800 Writing Program ]
NOTE: AM29L800BB needs 4 step sequences for 1 half-word data.
      So, the program time is twice of Starata flash<2 step sequences>.
[ Check AM29L800 ]
Manufacture ID= 1<0x0001>, Device ID<0x225B>=225b

Image Size:0h~3950h

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:0

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 1300 140
0 1500 1600 1700 1800 1900 1a00 1b00 1c00 1d00 1e00 1f00 2000 2100 2200 2300 2400 250
0 2600 2700 2800 2900 2a00 2b00 2c00 2d00 2e00 2f00 3000 3100 3200 3300 3400 3500 36
0 3700 3800 3900
End of the sector data writing!!!

E:\work\product\WARMW2413\program\sjf\W2413Dongle\W2413dongle\Debug>

```