

# **Your Atmel Board May Not Be Dead**

**Author:**

**James P. Lynch  
Grand Island, New York, USA  
February 1, 2008**

# INTRODUCTION

Anyone working with Olimex SAM7 boards has probably encountered this nightmare: after things have been working just great, suddenly nothing works! The JTAG fails, the flash downloads do nothing, the board appears dead. That's right, you have "bricked" your board and all is lost.

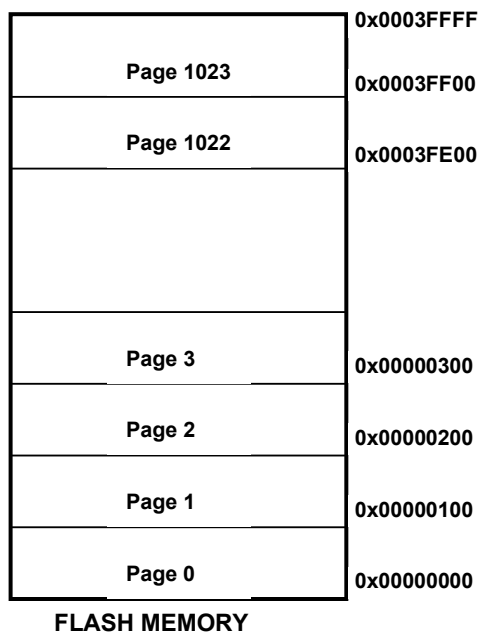
Before you take the "long swim" or cut another order to Spark Fun Electronics, there's a few simple checks you should do before giving up. Basically, there are three culprits that may be preventing your board from working.

- Lock Bits
- Security Bit
- Boot from Flash/ROM bit (GPNVM2)

I'll demonstrate some simple procedures to ensure that these bits are in their proper state. This may very well resurrect your board!

# LOCK BITS

The Atmel AT91SAM7S256 has 256k of FLASH memory organized as 1024 pages of 256 bytes/page.



The AT91SAM7S256 memory is further subdivided into "LOCK regions". There are 16 LOCK regions and each comprises 64 pages (16384 bytes per lock region). You cannot erase or program any of the data in a locked region. This will prevent an errant program from accidentally disturbing working firmware. It will also prevent OpenOCD from downloading into a "locked" region. OpenOCD will usually give an indication that it encountered a locked region.

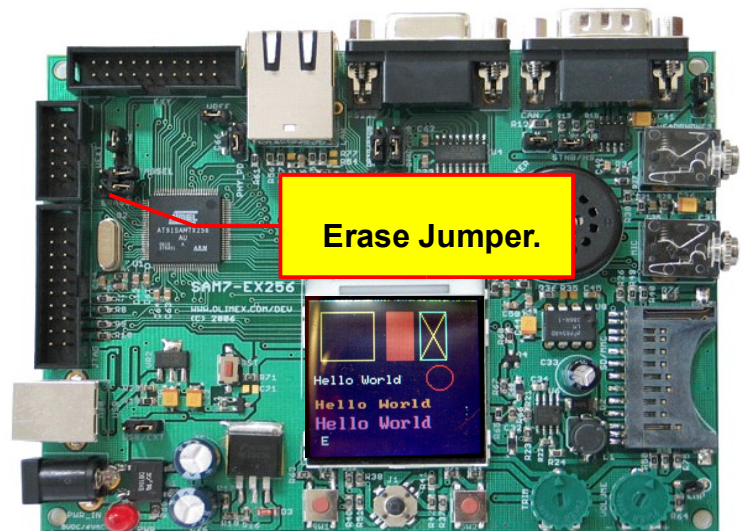
Page 1023		0x0003FFFF
Page 960	Lock Bit 63	0x0003C000
Page 959		
Page 896	Lock Bit 62	0x00038000
Page 256		
Page 255		
Page 192	Lock Bit 3	0x0000C000
Page 191		
Page 128	Lock Bit 2	0x00008000
Page 127		
Page 64	Lock Bit 1	0x00004000
Page 63		
Page 0	Lock Bit 0	0x00000000

### FLASH MEMORY

The LOCK bits can be set or cleared using the MC Flash Command Register (MC\_FCR) at address 0xFFFFF60.

There are several ways the lock bits can be set accidentally, either by software failure or improper use of the debugger. The OpenOCD debugger could be used to clear the lock bits using a memory write command, but there is a much easier way - **ERASE THE BOARD!** Here's the very simple procedure.

- First remove power.
- Install the “**Erase**” jumper.
- Apply power for 30 seconds.
- Your board gets erased and the lock bits are cleared.
- Remove power.
- Remove the “**Erase**” jumper.
- Re-apply power.



## Security Bit

The security bit is the most devious of all. If accidentally set, it will even prevent the JTAG from working. This is actually a very good idea. If you are afraid of competitors using the JTAG or the USB to read out your firmware in flash and reverse-engineer it, the “security” bit provides the fire wall. The security bit prevents any access off-chip to the flash contents (via JTAG, USB driver, etc.). Once set, the only way to clear the “security” bit is to **erase the flash** using the hardware “**erase**” jumper. That effectively locks out the prying eyes of competitors. Setting of the “security” bit is usually placed into the start-up software. Of course, an errant pointer or goofy use of the JTAG debugger can accidentally set this bit, making it look like the board is dead.

Use the “**ERASE**” procedure shown above to clear the security bit.

## Boot from Flash/ROM bit (GPNVM2)

When your Atmel SAM7 board boots, it will either run your code in flash memory or it will copy a boot loader program from hidden onchip ROM into flash so that the SAM-BA utility (USB-version typically) will run instead. The decision of which will run (flash or boot loader) is based on the General-purpose NVN Bit 2 (GPNVM2).

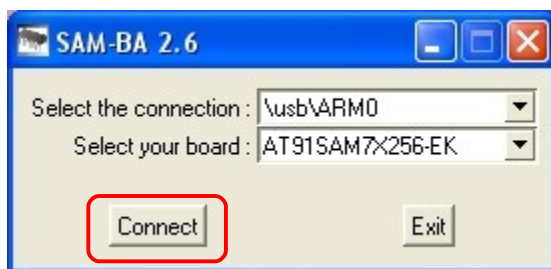
GPNVM2 = 0	Boot from hidden ROM (SAM-BA Boot Loader)
GPNVM2 = 1	Boot from FLASH

When you erase the board using the “**erase**” jumper procedure just described, it will **clear** the GPNVM2 bit which will direct the AT91SAM7X256 to boot the SAM-BA boot loader. The next time you plug in the USB cable and apply power, the onboard boot loader will run and you will hear the familiar USB “beep”. Now you can start the Atmel **SAM-BA** utility and use it to download your application. The act of downloading the application erases the boot loader and sets the GPNVM2 bit.

Consider what would happen if you use the hardware “erase” jumper and erase the entire flash. If you forget to run the **SAM-BA** which will set the bit when finished, you will leave the GPNVM bit cleared and only **SAM-BA** will run. This will cause JTAG downloaded applications to refuse to execute!

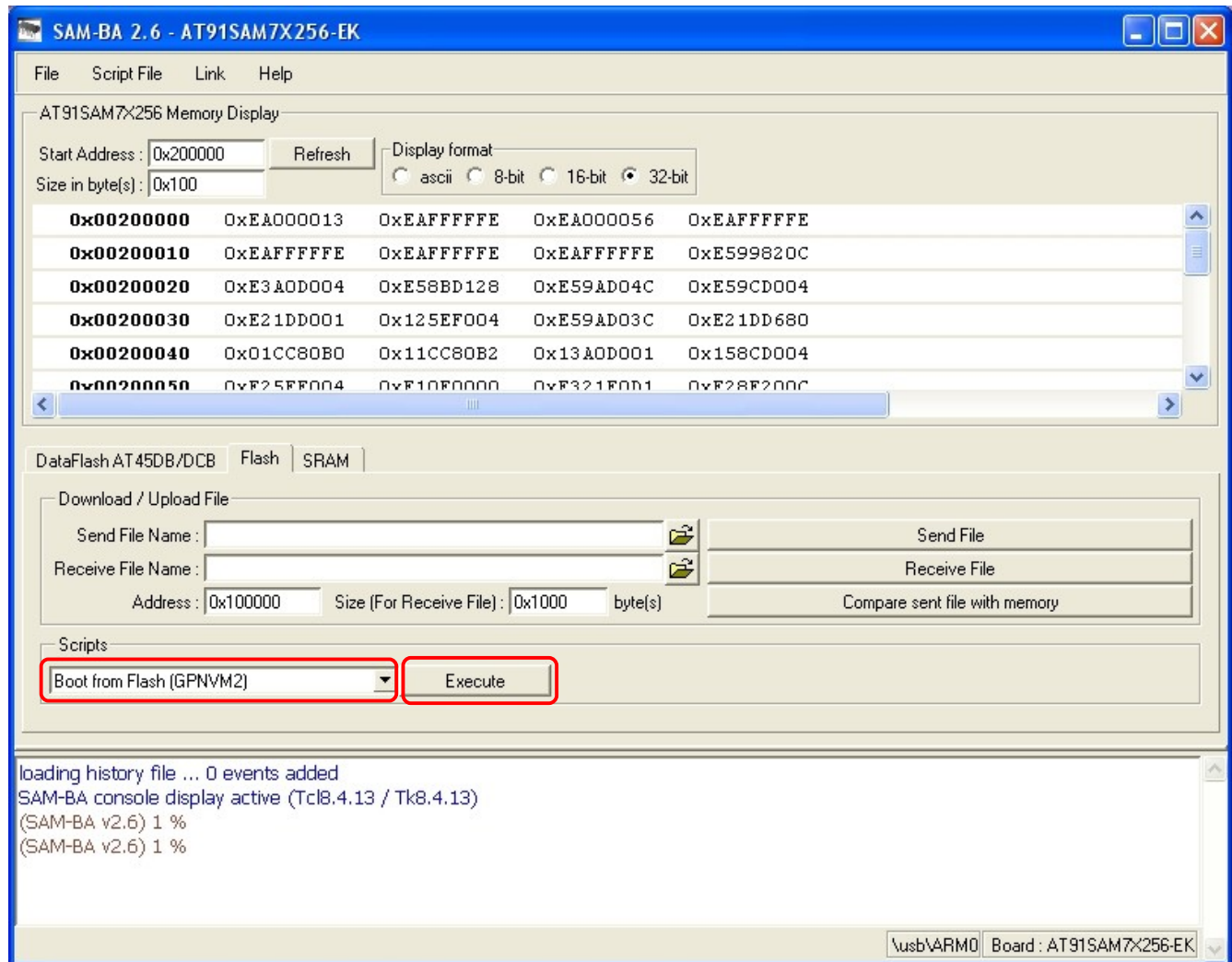
If you suspect that this may have happened, erase the board using the procedures above. Now install the USB cable and apply power, you should hear the familiar USB “beep”.

Start the Atmel **SAM-BA** application. The first pop-up window should show the following:



Select the USB connection and locate your board from the list of Atmel boards. Click “**Connect**” to start the Atmel **SAM-BA** utility.

The **SAM-BA** utility will present the following screen.



In the scripts shown at the bottom of the window, select “Boot from Flash (GPNVM2)” and hit the “Execute” button. This will set the GPNVM2 bit to boot from flash. Now exit from the SAM-BA utility and hook your JTAG debugger to your board. There’s a very good chance that now your board will work!

## Final Observations

When you have difficulty with your Atmel SAM7 board and you’re suspecting that it’s hardware trouble, make it a habit to run these simple procedures to clear the lock bits, the security bit and set the “boot from flash” bit.

It is always a good idea to have a simple embedded software “Hello World” application available to download to make sure that the hardware is still functional.

As a final resort, turn everything off and go to bed. You would be amazed how a clear head in the morning can spot blunders that were invisible the night before.