How to Install DM9000B Driver on S3C6400 Platform

This report is for the following hardware setting. If your hardware setting is not the same, you should modify the **"DM9000_BASE address"** and **"interrupt number"**. Please refer to S3C6400 datasheet for the detail.

- DM9000B CS# pin：nCS1 (bank 1)
- DM9000B INT pin：XEINT4 (high level)

The Bootloader and Linux version：

- Bootloader：u-boot-1.3.4-samsung_rel-1-2-0_20090507
- Linux：samsung-ap- 2.6_v2.6.28.6-s3c64xx-r2_090309

➢ U-BOOT (1.3.4):

DM9000B base address：**0x1800 0000**

1. Disable NET_MULTI, SMC911X, CS8900 and define DM9000B base address in include/configs/smdk6410.h

```
//#define CONFIG_NET_MULTI          ←mark this line

#undef CONFIG_DRIVER_SMC911X       ←note

//#define CONFIG_DRIVER_CS8900      ←mark this line

#define CS8900_BASE                0x18800300
#define CS8900_BUS16

#define CONFIG_DRIVER_DM9000       1
#define DM9000_BASE                0x18000000
```

2. Add the following in lib_arm/board.c

```
#ifdef CONFIG_DRIVER_DM9000
#include "../drivers/net/dm9000.h"
#endif

#ifdef CONFIG_DRIVER_DM9000
extern void DM9000_get_enetaddr (uchar * addr);
```

```
#endif

#ifdef CONFIG_DRIVER_DM9000
      DM9000_get_enetaddr (gd->bd->bi_enetaddr);
#endif
```

3. Copy dm9000.c, dm9000.h to drivers/net/ directory.
   and modify Makefile as below:

```
#COBJS-$(CONFIG_DRIVER_DM9000) += dm9000x.o
COBJS-$(CONFIG_DRIVER_DM9000) += dm9000.o
```

P.S. If you want to change Bank1 timing, you can modify board/samsung/smdk6410/smdk6410.c

```
#define CS8900_Tacs     (0x0)   // 0clk     address set-up
#define CS8900_Tcos     (0x4)   // 4clk     chip selection set-up
#define CS8900_Tacc     (0xE)   // 14clk    access cycle
#define CS8900_Tcoh     (0x1)   // 1clk     chip selection hold
#define CS8900_Tah      (0x4)   // 4clk     address holding time
#define CS8900_Tacp     (0x6)   // 6clk     page mode access cycle
```

Another set as a reference setting: Tacs (0), Tcos (1), Tacc(5), Tcoh(1), Tah(4), Tacp(6)

➢ Linux (2.6.28.6)

DM9000B base address：**0x1800 0000**

DM9000B INT pin：**XEINT4 (high level)**

1. Copy dm9000.c and dm9000.h to drivers/net/ directory

Check driver/net/Makefile for below:

```
obj-$(CONFIG_DM9000) += dm9000.o
```

At Samsung-ap-2.6 <dir> run "make smdk6410mtd_defconfig"

Note: Set the values to ./config first.

At Samsung-ap-2.6 <dir> run "make menuconfig"
Select [*] Nwtworking support
Enter Device Drivers
Select [*] Network device support
Select [*] Ethernet (10 or 100Mbit)
Select <*> DM9000 support

Check drivers/net/Kconfig file to find below:

```
config DM9000
        tristate "DM9000 support"
        depends on ARM || BLACKFIN || MIPS
        select CRC32
        select MII
        ---help---
           Support for DM9000 chipset.

           To compile this driver as a module, choose M here.    The module
           will be called dm9000.
```

2. Add arch/arm/mach-s3c6410/mach-smdk6410.c

Add include header file:

```
#include <plat/gpio-cfg.h> // NEW-ADD
#include <plat/regs-gpio.h> // NEW-ADD
```

In struct platform_device *smdk6410_devices[] __initdata =

```
//      &s3c_device_smc911x,
        &s3c_device_dm9000,
```

Add new function: (P.S. If you want to change Bank1 timing, you can modify __raw_writel to 'S3C64XX_SROM_BC1')

```
static void __init smdk6410_dm9000_set(void)
{
        unsigned int data, tmp;

        tmp = __raw_readl(S3C64XX_SROM_BW);
        tmp &= ~(S3C64XX_SROM_BW_WAIT_ENABLE1_MASK |
```

```
S3C64XX_SROM_BW_WAIT_ENABLE1_MASK |
            S3C64XX_SROM_BW_DATA_WIDTH1_MASK);
      tmp |= S3C64XX_SROM_BW_DATA_WIDTH1_16BIT;
      __raw_writel(tmp, S3C64XX_SROM_BW);

      __raw_writel(
            S3C64XX_SROM_BCn_TACS(0) | S3C64XX_SROM_BCn_TCOS(1)
|
            S3C64XX_SROM_BCn_TACC(5) |
S3C64XX_SROM_BCn_TCOH(1) |
            S3C64XX_SROM_BCn_TCAH(4) | S3C64XX_SROM_BCn_TACP(6)
|
            S3C64XX_SROM_BCn_PMC_NORMAL,
                  S3C64XX_SROM_BC1);

// ***
      //      s3c_gpio_cfgpin( S3C64XX_GPN(4),2); // Set EINT4 to be Ext.
Interrupt
      data = __raw_readl(S3C64XX_GPNCON);
            printk(KERN_INFO " ... RD GPNCON DAT ... %x\n", data);

            data &= ~S3C64XX_GPN_CONMASK(4);
            data |= (0x2 << ((4) * 2));
      __raw_writel(data, S3C64XX_GPNCON);
            printk(KERN_INFO " ... WR GPNCON DAT ... %x\n", data);
// ***

// >>>
      data = __raw_readl(S3C64XX_GPNPUD);
            printk(KERN_INFO " ... RD GPNPUD DAT ... %x\n", data);

            data &= ~(0x03<<8);
            data |= 0x02 << 8; // Set Pull-Up Enable
      __raw_writel(data, S3C64XX_GPNPUD);
            printk(KERN_INFO " ... WR GPNPUD DAT ... %x\n", data);
// >>>
```

```
        data = __raw_readl(S3C64XX_EINT0CON0);
            printk(KERN_INFO " ... RD S3C64XX_EINT0CON0 DAT ... %x\n",
data);

            //data= writel(S3C64XX_EINT0CON0);
             tmp= data & 0x700;
             if (tmp==0x000 || tmp==0x200 || tmp==0x300)
                tmp= 0x100;
             else if (tmp==0x100 || tmp==0x400 || tmp==0x500)
                tmp= 0x000;
              data &= ~0x700;
              data |= tmp;    /* (*Note) */
        __raw_writel(data, S3C64XX_EINT0CON0);
            printk(KERN_INFO " ... WR S3C64XX_EINT0CON0 DAT ...
%x\n", data);
}

/* (*Note)*/ For EINT5,4, tmp= 0x100 for High level, the default is 0x000 which
            is level.

In function “void smdk6410_machine_init(void)”:
      //smdk6410_smc911x_set();
       smdk6410_dm9000_set();    // NEW-ADD
```

3. Add arch/arm/plat-s3c/include/plat/devs.h

Add extern:

```
extern struct platform_device s3c_device_dm9000;
```

4. Add arch/arm/plat-s3c64XX/devs.c

Add include:

```
#include <linux/dm9000.h>
```

Add declare and initial value:

```
static struct resource s3c_dm9000_resources[] = {
        [0] = {
```

```
                .start    = 0x18000000, //0xe3600000,
                .end      = 0x18000000+3, //0xe3600000+3,
                .flags    = IORESOURCE_MEM,
        },
        [1] = {
                .start    = 0x18000000+4, //0xe3600000+4,
                .end      = 0x18000000+7, //0xe3600000+7,
                .flags    = IORESOURCE_MEM,
        },
        [2] = {
                .start = IRQ_EINT(4),
                .end      = IRQ_EINT(4),
                .flags = IORESOURCE_IRQ |
                         IORESOURCE_IRQ_HIGHLEVEL,
          },
};


static struct dm9000_plat_data bast_dm9k_platdata = {
      .flags            = DM9000_PLATF_16BITONLY,
};

struct platform_device s3c_device_dm9000= {
        .name                 = "dm9000",
        .id                   =   -1,
        .num_resources    = ARRAY_SIZE(s3c_dm9000_resources),
        .resource           = s3c_dm9000_resources,
      .dev        = {
            .platform_data = &bast_dm9k_platdata,
      }
};
```