Linux3.0.8平台搭建移植文档——LAN9220网卡驱动移植

1）定义 lan9220网卡 platform_device 资源

#vi arch/arm/mach-s5pv210/mach-smdkv210.c

在 mach-smdkv210.c 文件 struct platform_device smdkv210_dm9000结构体定义的后面，添加资源如下：

```
.....
struct platform_device smdkv210_dm9000 = {
    .name           = "dm9000",
    .id         = -1,
    .num_resources      = ARRAY_SIZE(smdkv210_dm9000_resources),
    .resource = smdkv210_dm9000_resources,
    .dev        = {
        .platform_data = &smdkv210_dm9000_platdata,
    },
};
//add by sunplusedu start
static struct smsc911x_platform_config smsc911x_config = {
    .irq_polarity   = SMSC911X_IRQ_POLARITY_ACTIVE_LOW,
    .irq_type       = SMSC911X_IRQ_TYPE_PUSH_PULL,
    .flags          = SMSC911X_USE_32BIT,
    .phy_interface  = PHY_INTERFACE_MODE_MII,
    .mac = {0x00,0x09,0xc0,0xff,0xec,0x48},
};

static struct resource smsc911x_resources[] = {
  [0] = {
      .start = S5PV210_PA_SROM_BANK5,
      .end   = S5PV210_PA_SROM_BANK5 + 0xFF,
      .flags = IORESOURCE_MEM,
  },
  [1] = {
      .start = IRQ_EINT(9),
      .end   = IRQ_EINT(9),
      .flags = IORESOURCE_IRQ | IORESOURCE_IRQ_LOWLEVEL,
  }
};

struct platform_device device_smsc911x = {
  .name     = "smsc911x",
  .id       =  0,
  .num_resources    = ARRAY_SIZE(smsc911x_resources),
  .resource  = smsc911x_resources,
```

```
        .dev        = {
            .platform_data = &smsc911x_config,
        }
    };
    //add by sunplusedu end
```

在 mach-smdkv210.c 文件开始部分添加网卡911x regs gpio 的头文件，如下：

```
#include <linux/smsc911x.h>
#include <mach/regs-gpio.h>
```

2）添加 paltform_device 资源

在 mach-smdkv210.c 文件添加网卡资源到资源列表（smdkv210_devices）中，注意注释掉 smdkv210_dm9000 资源,如下：

```
static struct platform_device *smdkv210_devices[] __initdata = {
        ......
        //&smdkv210_dm9000,
        &device_smsc911x,
        ......
};
```

3）添加911x 网卡初始化函数

在 mach-smdkv210.c 文件的 smdkv210_dm9000_init()函数的后面,添加函数如下：

```
static void __init smdkv210_dm9000_init(void)
{
    unsigned int tmp;
    ......
}
static void __init smdk210_smsc911x_init(void)
    {
        unsigned int tmp;
        tmp = ((0<<28)|(4<<24)|(13<<16)|(1<<12)|(4<<8)|(6<<4)|(0<<0));
        //tmp = ((0xf<<28)|(0xf<<24)|(0x1f<<16)|(0xf<<12)|(0xf<<8)|(6<<4)|(0<<0));
        __raw_writel(tmp, (S5P_SROM_BW + 0x18));
        tmp = __raw_readl(S5P_SROM_BW);
        tmp &= ~(0xf << 20);
        tmp |= (0x3 << 20);

        __raw_writel(tmp, S5P_SROM_BW);
        tmp = __raw_readl(S5PV210_MP01_BASE);
        tmp &= ~(0xf << 20);
        tmp |= (2 << 20);
```

```
        __raw_writel(tmp, S5PV210_MP01_BASE);
    }
```

在 smdkv210_machine_init 函数中调用所添加函数，注释掉 smdkv210_dm9000_init 函数的调用，如下所示：

```
static void __init smdkv210_machine_init(void)
{
    s3c_pm_init();
   //smdkv210_dm9000_init();
    smdk210_smsc911x_init();
    ......
}
```

4）添加 MP01口虚拟地址资源

在 arch/arm/mach-s5pv210/include/mach/regs-gpio.h 文件后面，添加以下内容：

```
#define S5PV210_MP01_BASE          (S5P_VA_GPIO + 0x2E0)
```

5）make menuconfig 配置网卡驱动

执行 make menuconfig 命令后，选中 network support 选项：

```
[*] Networking support  --->
        Networking options  --->
              <*> Packet socket
              <*> Unix domain sockets
              •••
              [*] TCP/IP networking
              [ ]   IP: multicasting
              [ ]   IP: advanced router
              •••
```

回到最顶层，配置网卡设备驱动的支持：

```
Device Drivers  --->
      [*] Network device support  --->                //选中该选项并进入
            ...
            [*]   Ethernet (10 or 100Mbit)  --->  //选中10M 或 100M 网卡选项
                  <*> SMSC LAN911x/LAN921x families embedded ethernet support
            [ ]   Ethernet (1000 Mbit)  --->
            [ ]   Ethernet (10000 Mbit)  --->
            ...
```

6）make

将在arch/arm/boot/下生成编译好的可执行程序zImage下载到开发板即可，执行命令“ifconfig eth0”不再提示找不到设备，接下来可以设置IP等进行网络常规测试

注意：

支持nfs作为根文件系统配置

我们有时在调试文件系统时，需要远程网络启动根文件系统，前提是内核必须是支持的，所以我们需进行如下配置来让内核支持 nfs 根文件系统启动。

```
#cd linux-3.0.8
#make menuconfig
       [*] Networking support  --->
            Networking options  --->
              [*]   IP: kernel level autoconfiguration(若未选中，则选中此选项)

#make menuconfig
       File systems --->
           [*] Network File Systems  --->
               <*>   NFS client support
                 [*]     Root file system on NFS
               <*>   NFS server support(选中此选项)
```