

Linux 中如何把串口 0 改为普通串口使用

开发板自带的 Linux 系统中，串口 0（对应设备/dev/ttySAC0）已经被用作控制台使用，如何把它作为普通串口使用，需要改两个地方：

1. 在 bootloader 中把 Linux 启动参数的串口输出改为 null，根据使用 bootloader 的不同，可以又分为两种情况：(1)使用 supervivi 作为 bootloader (2)使用 vboot 作为 bootloader
2. 屏蔽内核解压时的打印信息

目录

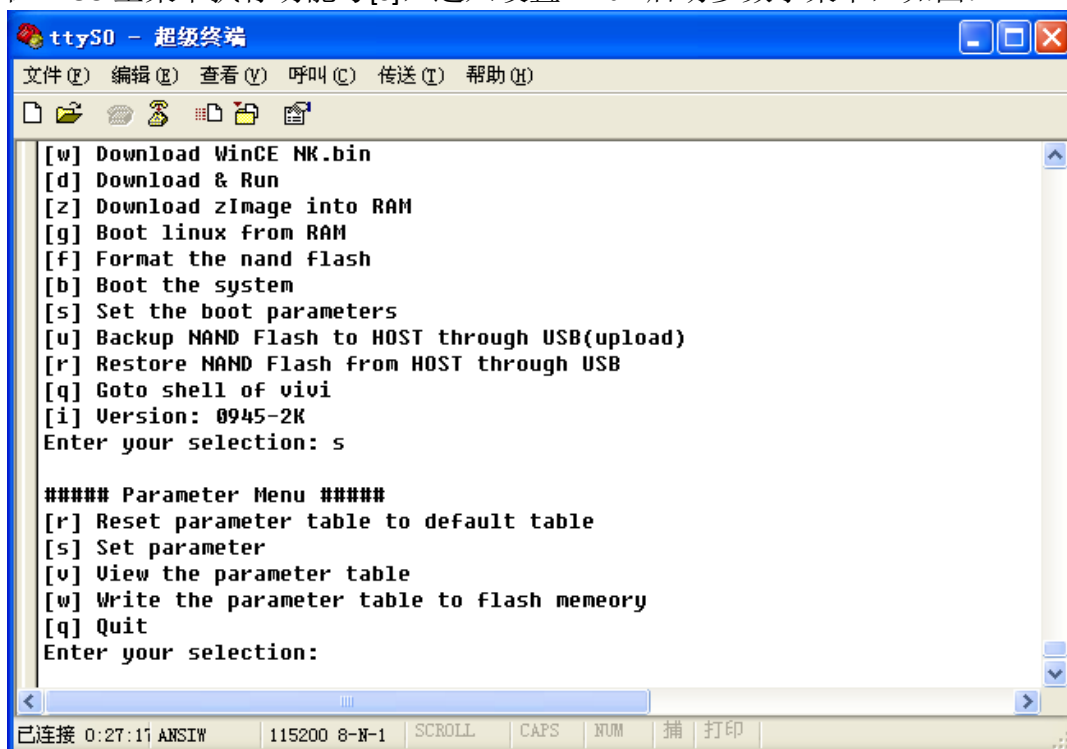
1. 在 bootloader 中修改启动内核启动参数	1
1.1 当使用 supervivi 作为 bootloader 时，修改 Linux 启动参数的方法	1
1.2 当使用 vboot 作为 bootloader 时，修改 Linux 启动参数的方法	4
2 屏蔽内核解压时的打印信息	5

下面是详细的修改步骤：

1. 在 bootloader 中修改启动内核启动参数

1.1 当使用 supervivi 作为 bootloader 时，修改 Linux 启动参数的方法

如果你使用了 supervivi 作为 bootloader，可以把内核启动时的输出信息屏蔽掉，或者改为其他串口输出，但无法去掉 supervivi 本身的输出信息。参考用户手册 2.2.4 章节的方法修改，如下：在 BIOS 主菜单执行功能号[s]，进入设置 Linux 启动参数子菜单，如图：



(1)浏览当前参数设置[v]

输入“v”可以浏览当前启动参数设置情况：

```
ttyS0 - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)

[w] Write the parameter table to flash memory
[q] Quit
Enter your selection: v
Number of parameters: 9
name                :          hex          integer
-----
mach_type            :      000007cf          1999
media_type           :      00000003           3
boot_mem_base        :      30000000      805306368
baudrate             :      0001c200      115200
xmodem               :      00000001           1
xmodem_one_nak       :      00000000           0
xmodem_initial_timeout :      000493e0      300000
xmodem_timeout       :      000f4240      1000000
boot_delay           :      01000000      16777216
Linux command line: noinitrd root=/dev/mtdblock2 init=/linuxrc console=ttySAC

##### Parameter Menu #####
[r] Reset parameter table to default table
[s] Set parameter
[v] View the parameter table
[w] Write the parameter table to flash memory
[q] Quit
Enter your selection: _
```

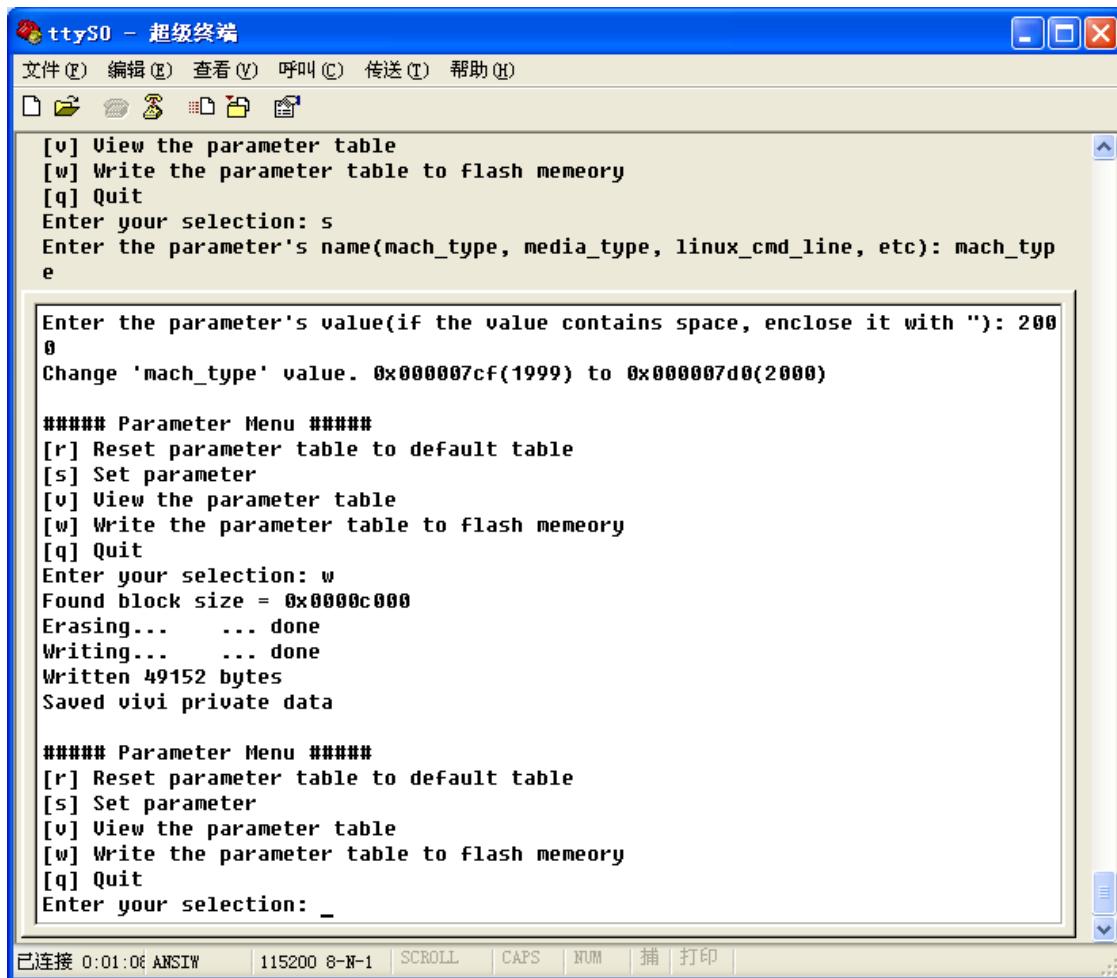
(2)设置参数[s]

输入“s”，可以对上面列出的参数进行设置，比较常用的参数有(其他参数建议不要更改)：

- Mach_type
- Linux command line

下面我们分别举例说明如何设置：

开发板默认的 MACH_TYPE 为 1999，假设你编译的内核使用的 MACH_TYPE 是 2000，则可以通过修改 mach_type 参数来正常启动内核，根据提示先输入参数的名字“mach_type”，再输入参数值“2000” (引号不要输入)，更改后记得输入“w”保存设置，如图：



```
[v] View the parameter table
[w] Write the parameter table to flash memeory
[q] Quit
Enter your selection: s
Enter the parameter's name(mach_type, media_type, linux_cmd_line, etc): mach_type
Enter the parameter's value(if the value contains space, enclose it with "): 2000
Change 'mach_type' value. 0x000007cf(1999) to 0x000007d0(2000)

##### Parameter Menu #####
[r] Reset parameter table to default table
[s] Set parameter
[v] View the parameter table
[w] Write the parameter table to flash memeory
[q] Quit
Enter your selection: w
Found block size = 0x0000c000
Erasing...    ... done
Writing...    ... done
Written 49152 bytes
Saved vivi private data

##### Parameter Menu #####
[r] Reset parameter table to default table
[s] Set parameter
[v] View the parameter table
[w] Write the parameter table to flash memeory
[q] Quit
Enter your selection: _
```

linux_cmd_line 是经常用到的一个内核启动参数，例如要把内核的启动信息和登录终端改为串口 1（默认是串口 0），则这样修改：

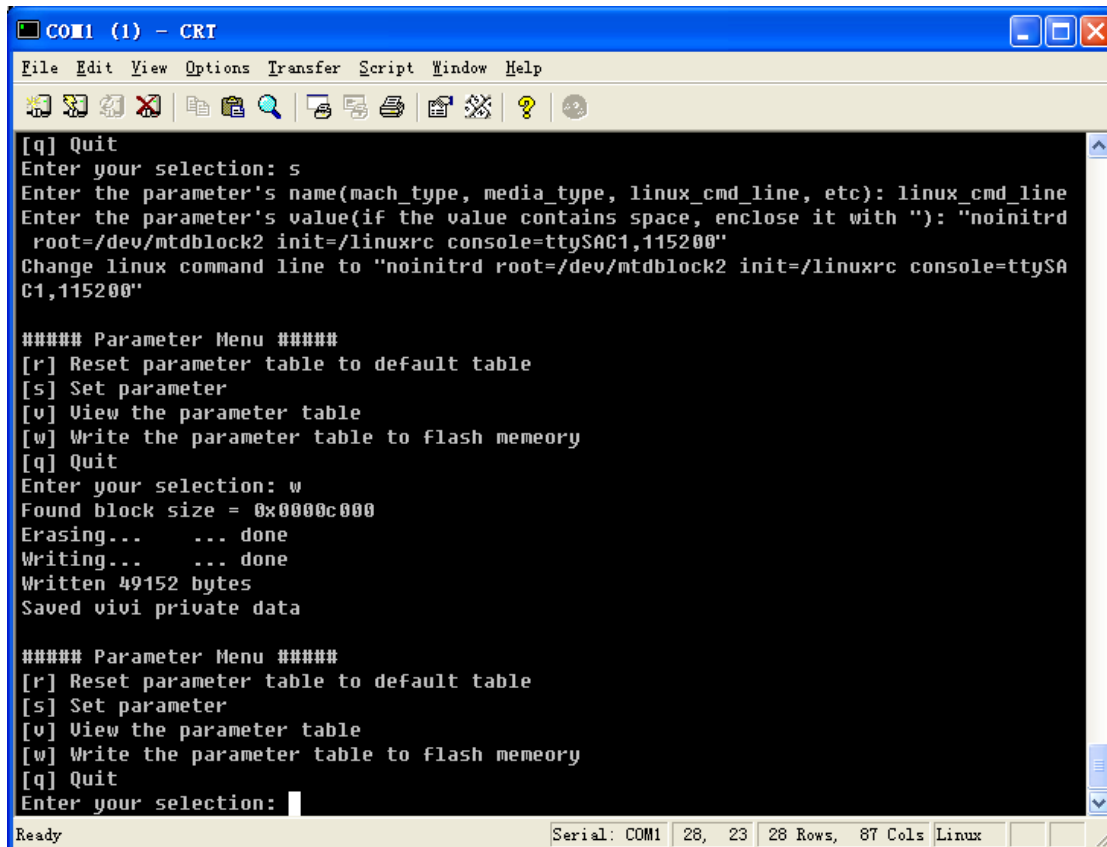
通过浏览参数，可以看到原来的参数：

linux_cmd_line: `noinitrd root=/dev/mtdblock2 init=/linuxrc console=ttySAC0`

输入“s”后，根据提示输入要修改的参数“linux_cmd_line”，回车，再输入参数值为(因为该参数串中有空格，因此需要输入双引号括起来)：

“noinitrd root=/dev/mtdblock2 init=/linuxrc console=ttySAC1,115200”

如图所示：



```
COM1 (1) - CRT
File Edit View Options Transfer Script Window Help

[q] Quit
Enter your selection: s
Enter the parameter's name(mach_type, media_type, linux_cmd_line, etc): linux_cmd_line
Enter the parameter's value(if the value contains space, enclose it with "): "noinitrd
root=/dev/mtdblock2 init=/linuxrc console=ttySAC1,115200"
Change linux command line to "noinitrd root=/dev/mtdblock2 init=/linuxrc console=ttySA
C1,115200"

##### Parameter Menu #####
[r] Reset parameter table to default table
[s] Set parameter
[v] View the parameter table
[w] Write the parameter table to flash memeory
[q] Quit
Enter your selection: w
Found block size = 0x0000c000
Erasing...    ... done
Writing...    ... done
Written 49152 bytes
Saved vivi private data

##### Parameter Menu #####
[r] Reset parameter table to default table
[s] Set parameter
[v] View the parameter table
[w] Write the parameter table to flash meneory
[q] Quit
Enter your selection: 
```

这样系统启动的时候，内核的启动信息和登录信息都将在串口 1 出现，而 supervivi 本身的输出信息不会改变，还是从串口 0 出来。

如果打算把内核信息屏蔽掉，可以把以上参数改为：

“noinitrd root=/dev/mtdblock2 init=/linuxrc console=null,115200”

(3)保存配置[w]

当设置更改之后，可以输入“w”保存所作的更改。

(4)恢复默认值[r]

输入“r”可以恢复出厂时的内核启动参数。

(5)返回主菜单[q]

输入“q”可以返回 BIOS 功能主菜单。

1.2 当使用 vboot 作为 bootloader 时，修改 Linux 启动参数的方法

使用 vboot 作为 bootloader，既可以方便的屏蔽掉 vboot 本身的串口输出信息，也可以修改内核启动时的输出信息，但需要重新编译 vboot 才可以，如下步骤：

(1) 先屏蔽 vboot 本身的输出信息

打开 vboot 源代码的 244x_lib.c 文件，找到 static inline void Uart_SendByte(int data)函数，添加如下红色部分的代码：

```
static inline void Uart_SendByte(int data)
{
    return ; //直接返回，不需要从串口输出任何东西
    while (!(UTRSTAT0 & 0x2)) ;    //Wait until THR is empty.
    UTXH0 = data;
}
```

这样可以屏蔽掉 vboot 本身的任何串口输出信息，比如“load Image of Linux...”等

(2) 再修改 vboot 中内核启动参数的设定

打开源代码 parameters.h 文件，把内核启动参数改为如下所示(蓝色为原有参数，红色为修

改后的参数):

```
//#define LINUX_CMD_LINE "noinitrd root=/dev/mtdblock3 init=/linuxrc console=ttySAC0"  
#define LINUX_CMD_LINE "noinitrd root=/dev/mtdblock3 init=/linuxrc console=null"
```

这样，我们就完成了 vboot 的修改，在命令行下输入: make，重新编译生成 vboot.bin，并使用新的 vboot.bin 启动系统就可以了(把开发板 S2 设置为 NOR 启动，通过 supervivi 的“v”功能可以烧写 vboot.bin 到 nand flash 的 block 0 起始的位置，以作为 bootloader)

2 屏蔽内核解压时的打印信息

通过以上修改，虽然可以更改内核启动时的大部分信息，但发现依然会有这样的信息从串口 0 冒出：

```
Uncompressing      Linux.....  
done, booting the kernel.
```

这里的信息其实是在内核中产生的，打开 linux-2.6.32.2/arch/arm/boot/compressed/misc.c 文件，找到这个函数；static void putstr(const char *ptr)，使用上面的同样方法，修改如下（红色部分）：

```
static void putstr(const char *ptr)  
{  
    char c;  
  
    return ; //直接返回，不作任何输出  
    while ((c = *ptr++) != '\0') {  
        if (c == '\n')  
            putc('\r');  
        putc(c);  
    }  
  
    flush();  
}
```

修改后，重新编译内核，并把它烧写到开发板，可以看到串口 0 不再有任何输出了。