

9200 核心板 DBGU 测试

——Team Mcuzone

所谓 9200 核心板的 DBGU 测试就是通过 9200 的 DBGU 接口对芯片进行 RAM 或者 FLASH 的烧写操作。通过该操作验证 9200 是否正确并稳定运行。

首先进行的是测试是片内的启动程序的下载调试。在开始前请检查电源链接，**FLASHBOOT 跳线需要悬空**。然后找一个串口模块，这里用的是 ARK3116 模块。请把 ARK3116 模块上的 TXD 脚连接到 9200 EK 核心板的 PA31，将 ARK3116 模块上的 RXD 脚链接到 9200 EK 核心板的 PA30 脚。然后把核心板的 GND 和 ARK3116 模块的 GND 连接起来，实现共地。

9200 EK 核心板上电，打开串口调试软件，如 SSCOM，设置成 115200 波特率，8 位数据位，1 位停止位，无硬件流控制。如果设置正确，应该会接收到字符 C，表示 9200 片内启动成功，等待来自串口端的数据。



图 1、SSCOM 接收字符 C

也可以使用 WINDOWS 自带的超级终端来进行接收，超级终端默认在：程序->附件->通讯，打开后需要进行一点设置：



图 2，超级终端下新建连接
新建一个连接，如图 2。

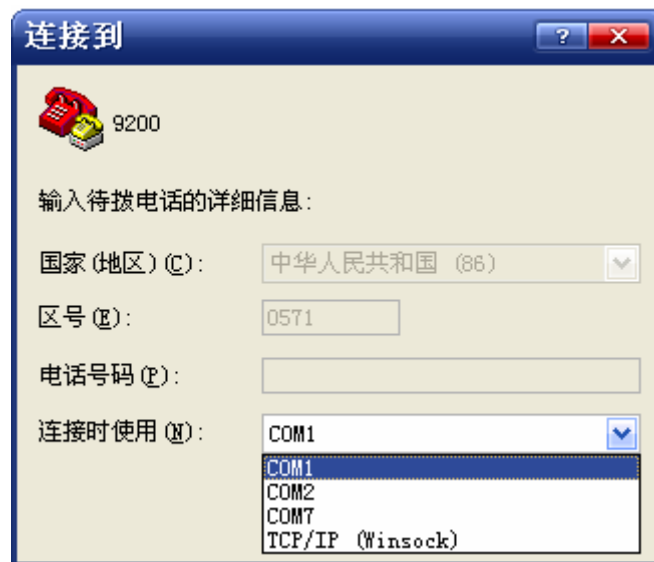


图 3，选择串口
选择要使用的串口，如图 3。

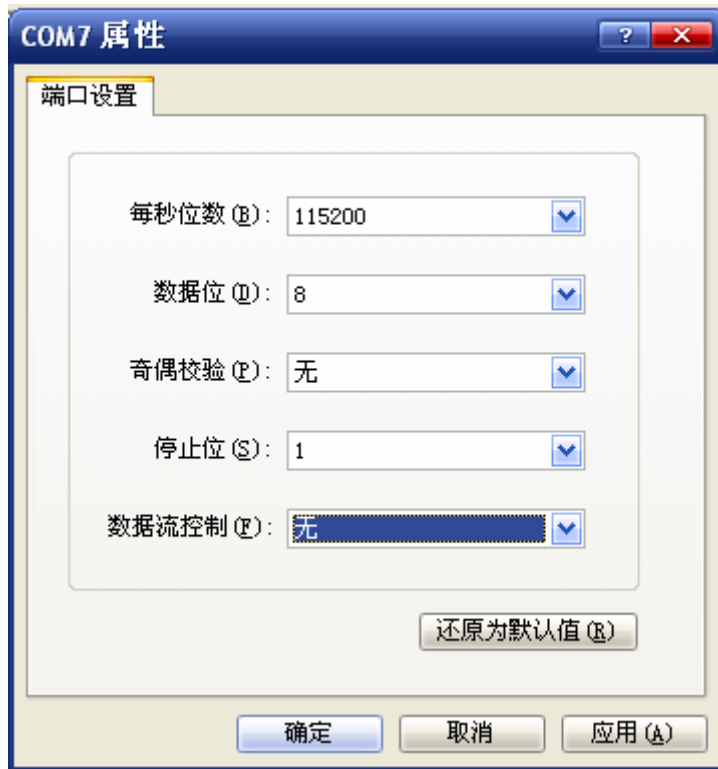


图 4，串口设置

对串口进行设置，如图 4。

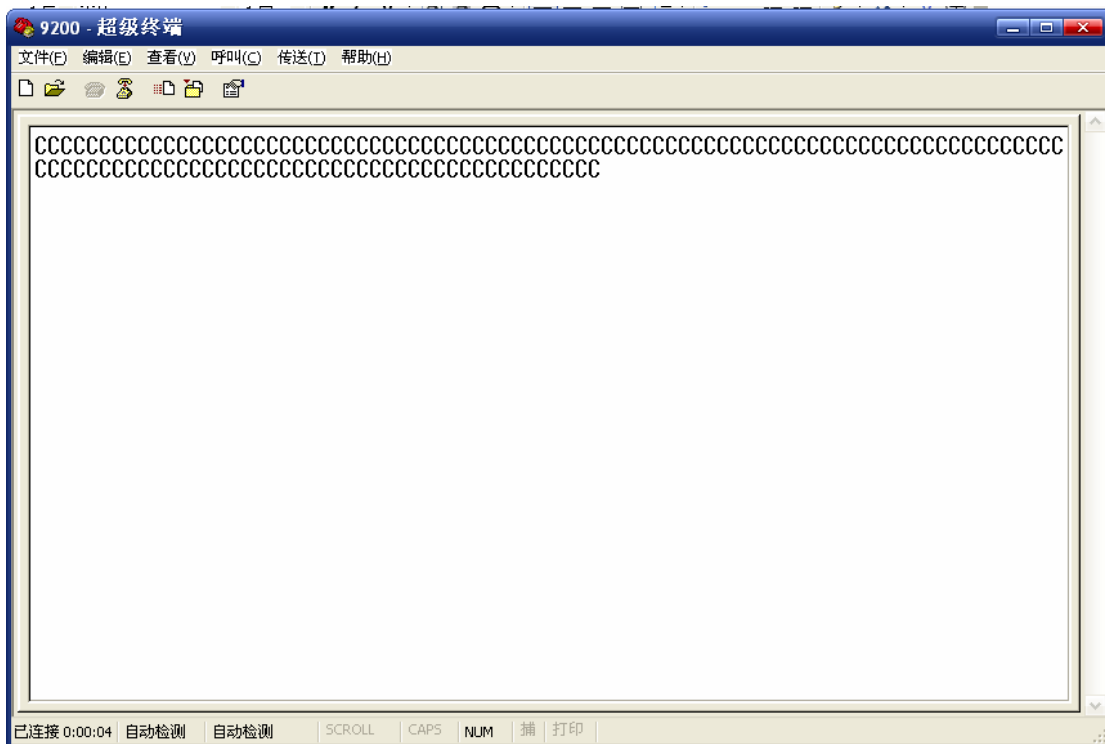


图 5，接收到字符 C

设置成功后，应该可以接收到字符 C。

看到字符 C 后，我们可以通过 XMODEM 协议将程序下载到目标板上，我

们首先下载一个很小的测试程序。

注意，其他的串口调试软件虽然可以正确接收数据，很多也能发送文件，但是基本上都不能用 Xmodem 的协议发送文件！所以，推荐直接使用超级终端来接收和发送。

首先，在连接成功的情况下，在“传送”菜单里面选择“发送文件”。如下图 6。

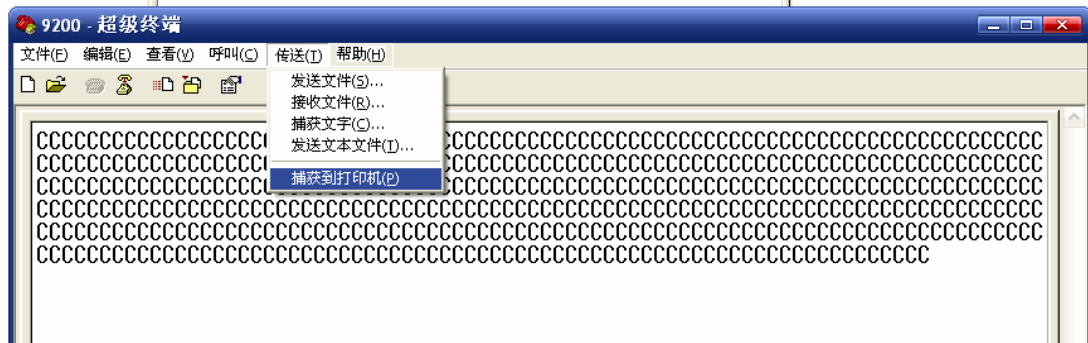


图 6，选择“发送文件”

然后超级终端会弹出一个发送文件的对话框，如下图 7，请找到需要发送的 bin 文件，这里我们使用的是一个已经编译好的 ADS 的范例，是在片内 RAM 运行的测试程序。该测试程序可以到我们的网站，www.mcuzone.com 进行下载。

注意，协议要选择成 Xmodem！如下图 7 所示。



图 7，发送文件设置

选择好文件，选好协议后，点击“发送”按钮即开始进行文件发送，图 8 是发送截图，由于文件较小，仅为 4K，在 115200 波特率下 1-2 秒即可完成下载。



图 8，发送过程截图

发送成功后，超级终端即可接受到相应提示，如下图，与下载下去的程序预计输出内容完全一致，说明程序已经正常运行。

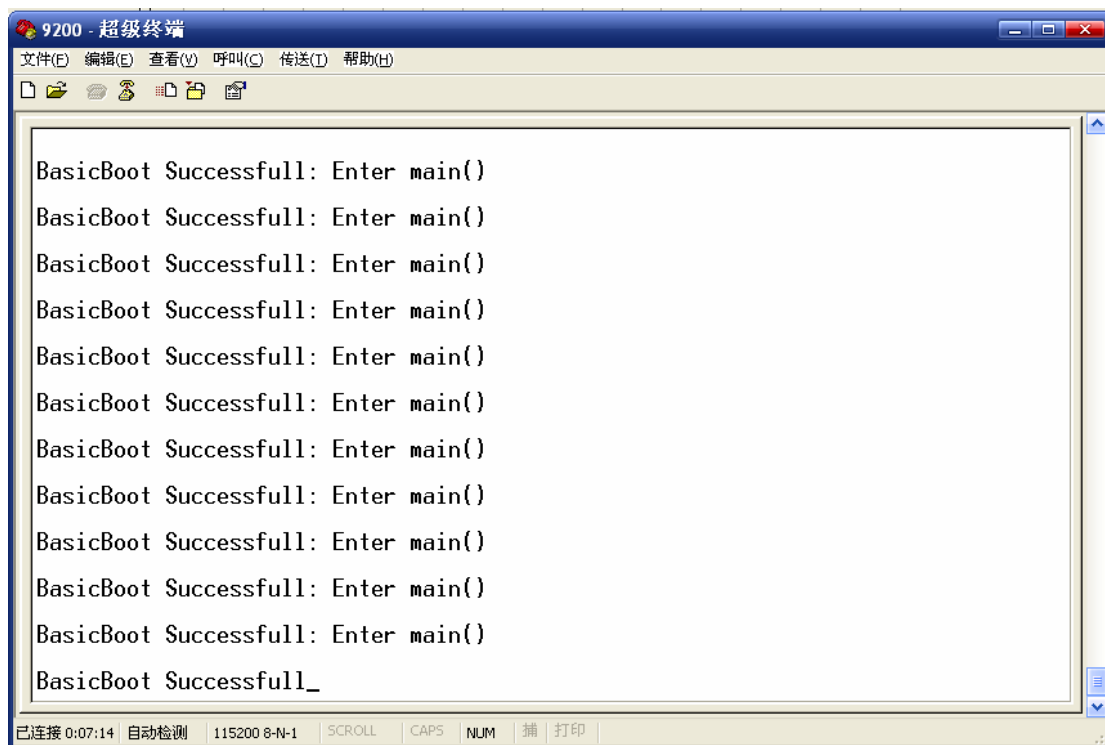
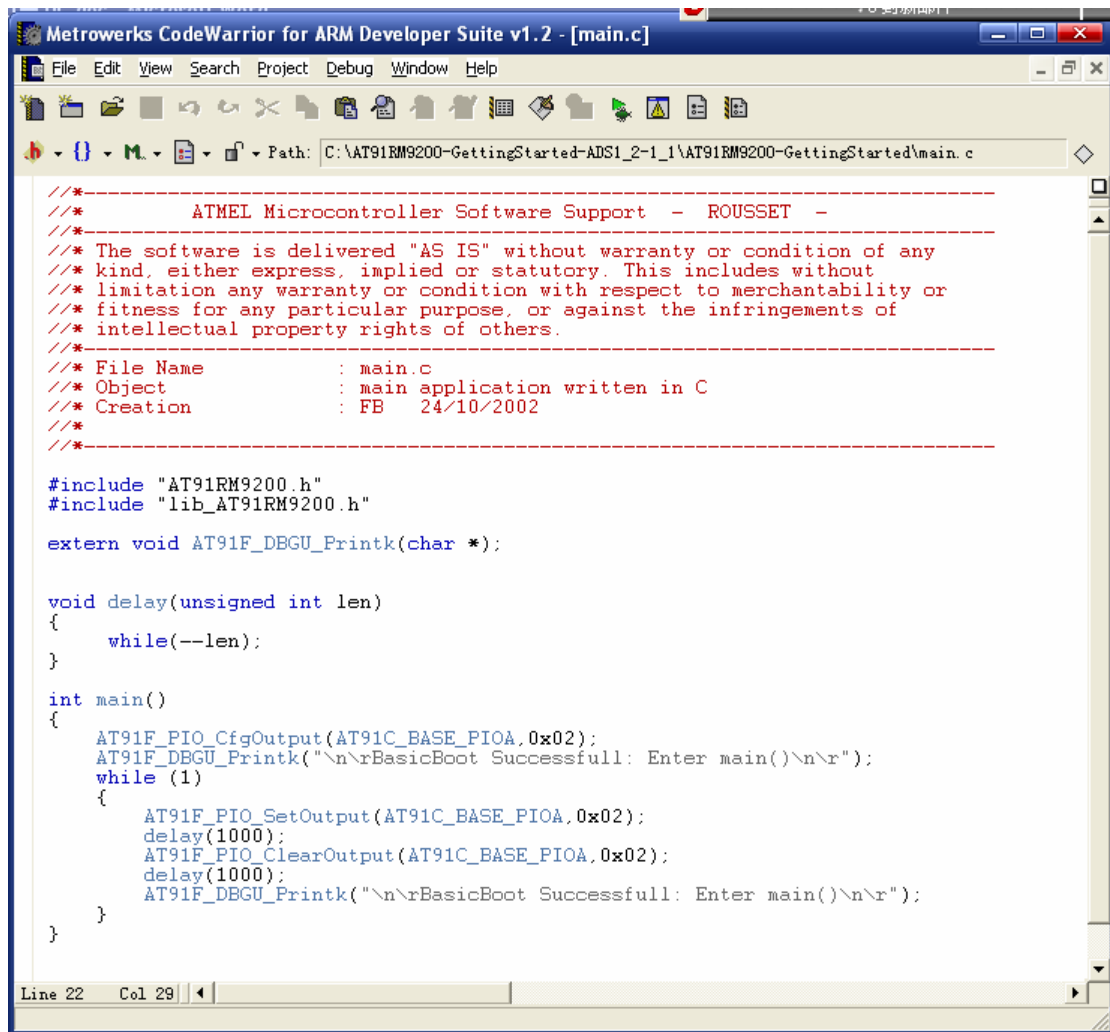


图 9，运行结果反馈



```
Metrowerks CodeWarrior for ARM Developer Suite v1.2 - [main.c]
File Edit View Search Project Debug Window Help
Path: C:\AT91RM9200-GettingStarted-ADS1_2-1_1\AT91RM9200-GettingStarted\main.c

/**-----
/**          ATMEL Microcontroller Software Support  -  ROUSSET  -
/**-----
/** The software is delivered "AS IS" without warranty or condition of any
/** kind, either express, implied or statutory. This includes without
/** limitation any warranty or condition with respect to merchantability or
/** fitness for any particular purpose, or against the infringements of
/** intellectual property rights of others.
/**-----
/** File Name      : main.c
/** Object         : main application written in C
/** Creation       : FB   24/10/2002
/**-----
/**-----

#include "AT91RM9200.h"
#include "lib_AT91RM9200.h"

extern void AT91F_DBGU_Printk(char *);

void delay(unsigned int len)
{
    while(--len);
}

int main()
{
    AT91F_PIO_CfgOutput(AT91C_BASE_PIOA, 0x02);
    AT91F_DBGU_Printk("\n\rBasicBoot Successfull: Enter main()\n\r");
    while (1)
    {
        AT91F_PIO_SetOutput(AT91C_BASE_PIOA, 0x02);
        delay(1000);
        AT91F_PIO_ClearOutput(AT91C_BASE_PIOA, 0x02);
        delay(1000);
        AT91F_DBGU_Printk("\n\rBasicBoot Successfull: Enter main()\n\r");
    }
}

Line 22 Col 29
```

图 10, 主程序代码

简单的接收，下载操作基本就是这样。

退出超级终端的时候，系统会提示是否需要保存当前连接，建议保存，下次可以直接打开使用，无需再次设置。

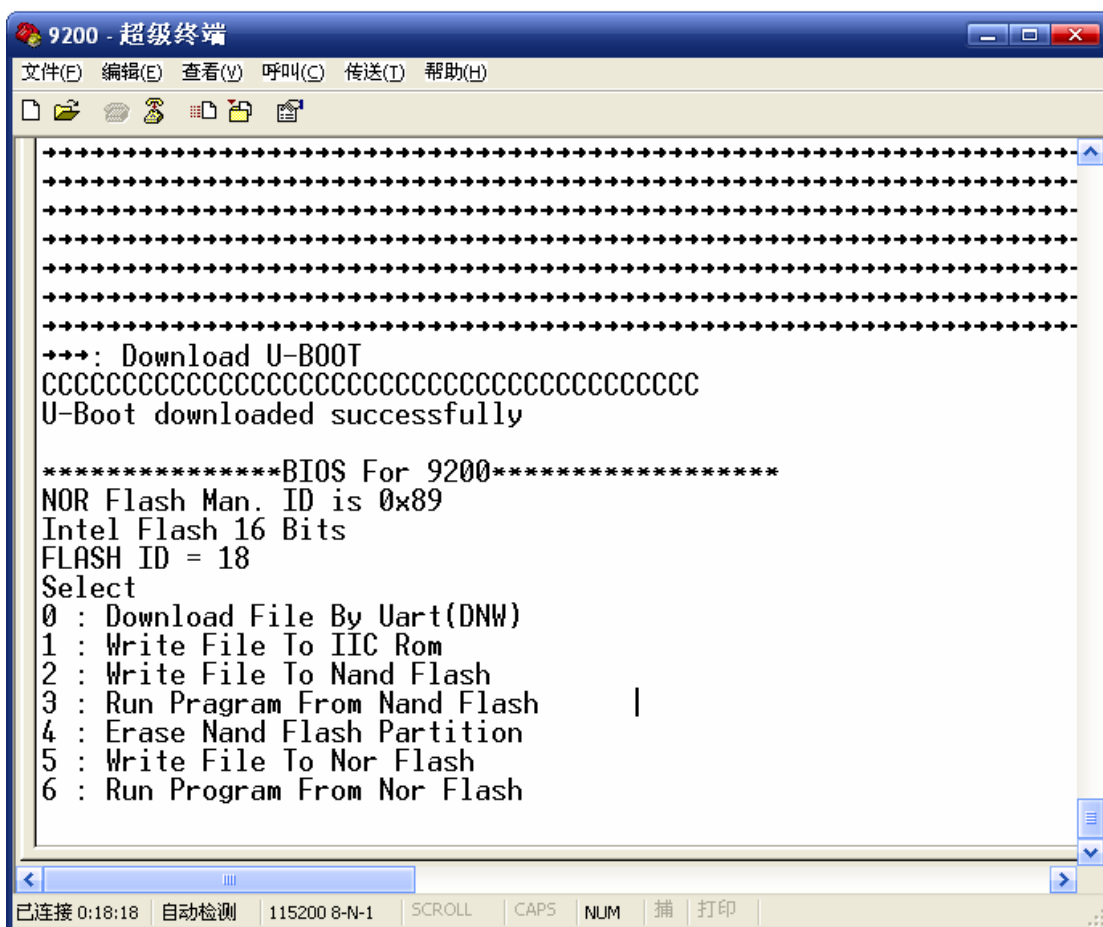


图 12, 下载 bios9200.bin

从输出的信息来看, bios9200 已经检测到 NOR FLASH 的型号, 和实际 28F128 的 ID 相符。

输出信息里面一共有 7 个操作可以选择, 由于我们的核心板没有 IIC ROM (EEPROM/24XX), 也没有 NAND FLASH, 所以, 我们只能选择 0, 5 和 6 三个操作。

下面用串口工具 DNW 来进行下一步操作。

首先运行 DNW 软件, 如下图, 没有任何内容, 需要进行一些设置。

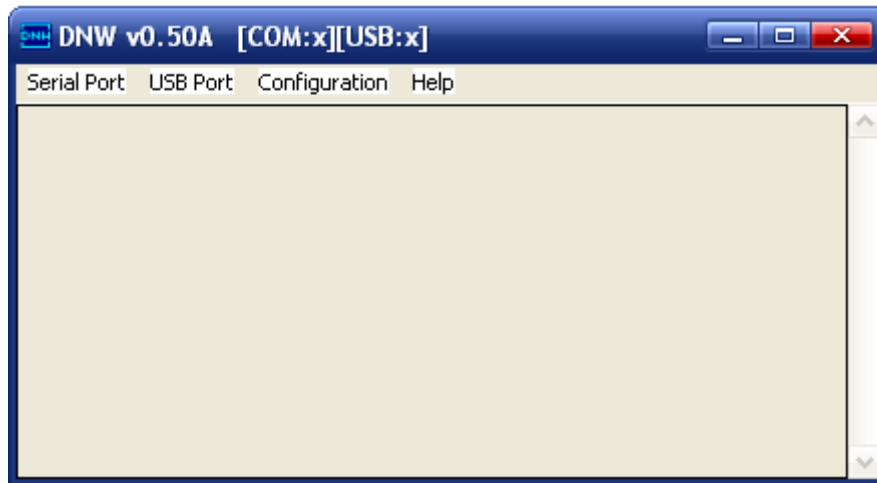


图 13, 打开 DNW

点击“Configuration”下的“Options”

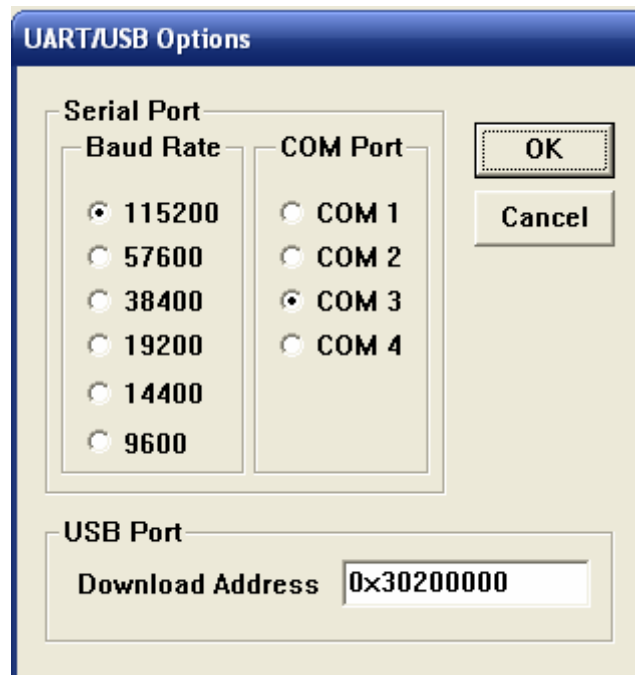


图 14, UART 设置

设置完成点击 OK。

然后再到“Serial Port”菜单选择“Connect”子菜单，**注意，要按一下回车，才会出现下图内容。**



图 15, 连接 9200

然后选择从 UART 下载, 即从 DBGU 接口下载, 如图 16。

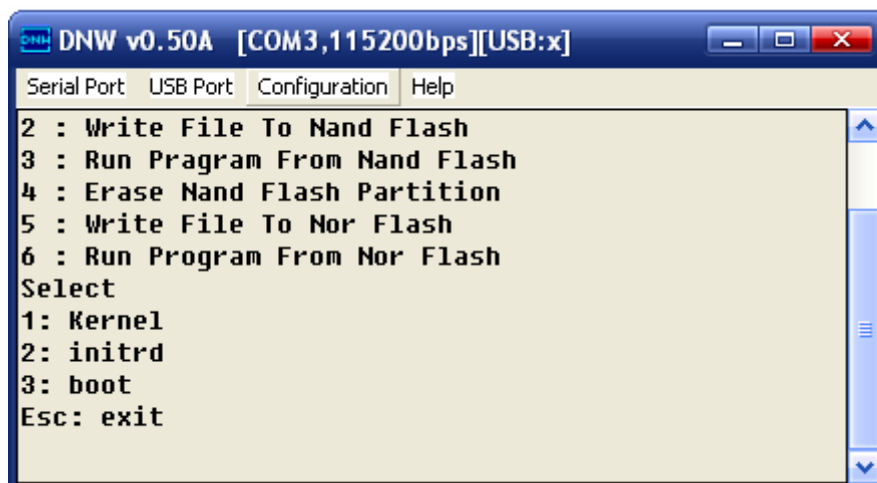


图 16, 选择 0

然后再选择 2, 如图 17。

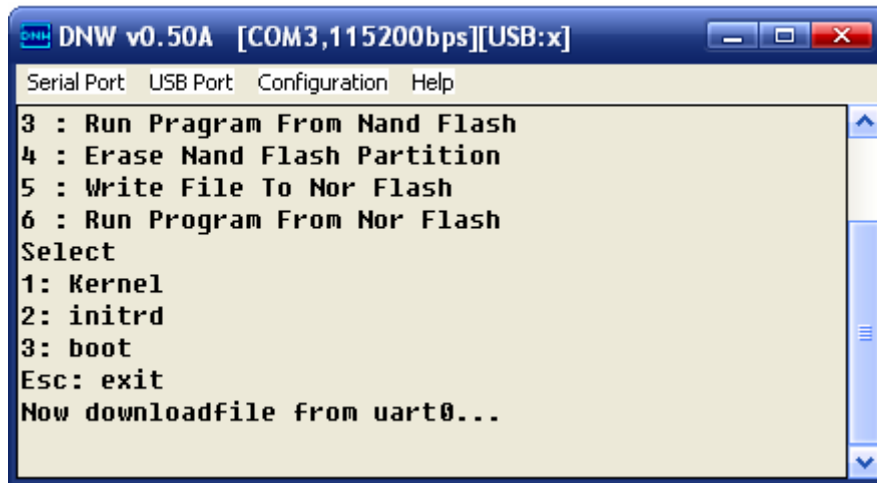


图 17, 选择 2

在出现“Now downloadfile from uart0”后，在“Serial Port”菜单下选择“Transmit”，然后出现如下图 18 的对话框：

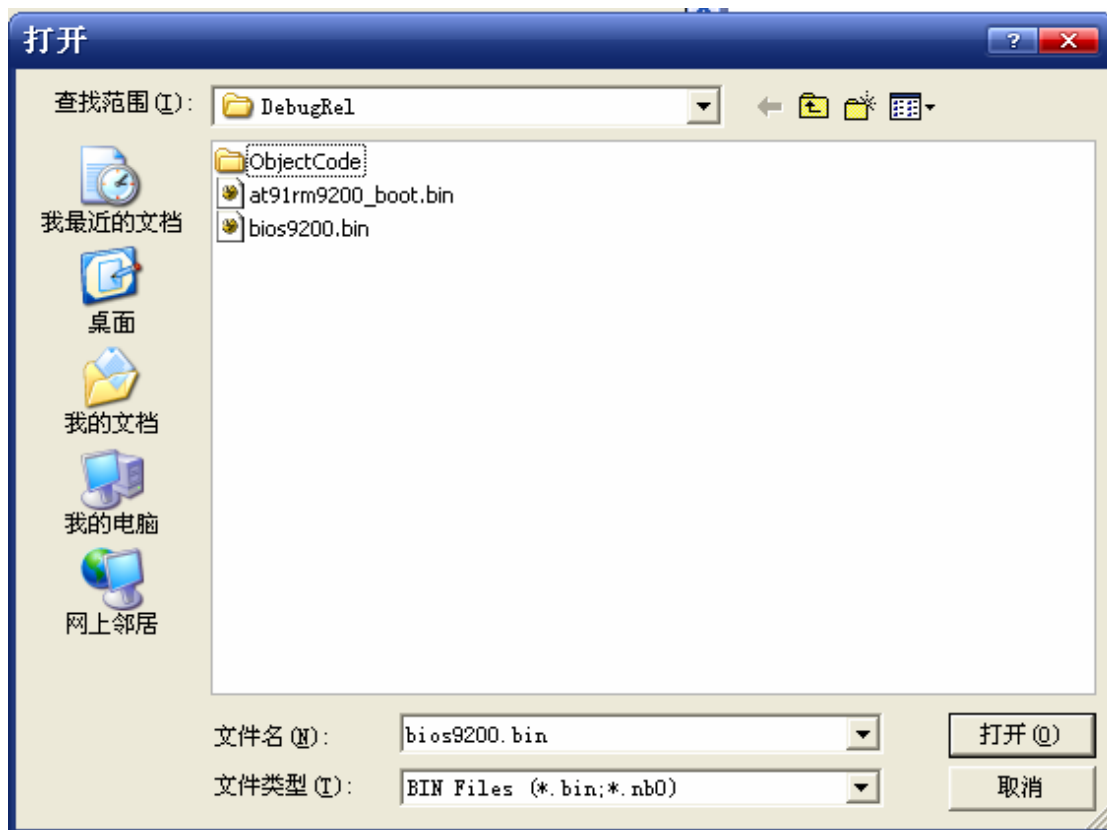


图 18, 下载 bios9200.bin

还是选择 bios9200.bin 文件。

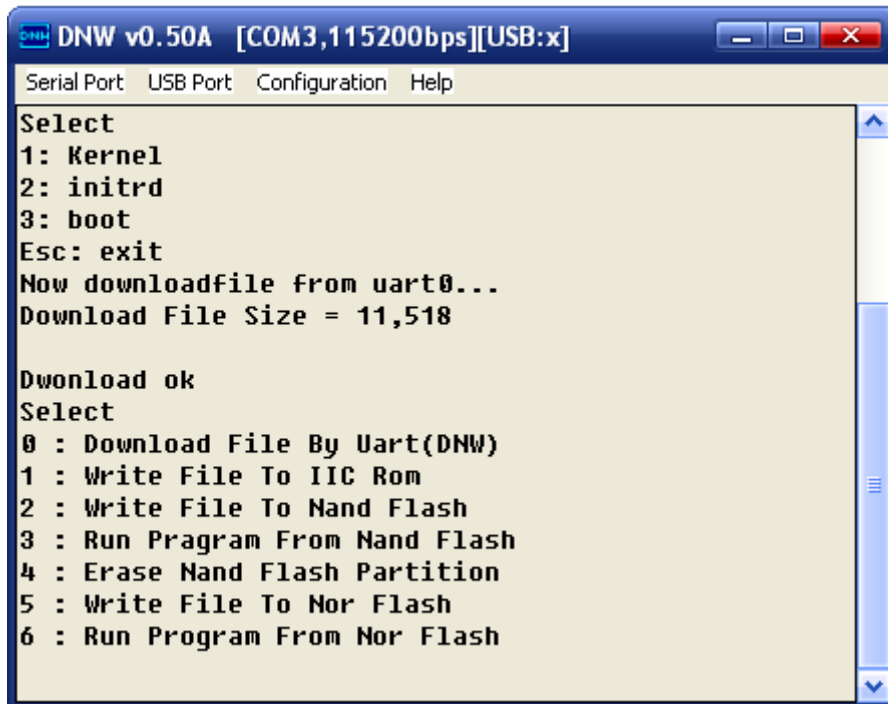


图 19, 选择 5

点击确认后, 很快就下载完成, 下载完成后出现图 19 所示信息。

由于我们没有 EEPROM, 也没有 NAND FLASH, 所以我们只能选择和 NOR FLASH 相关的操作进行测试。我们先选择 5, 即把刚才下载的 bios9200.bin 烧写到 NOR FLASH。运行信息如下图所示:

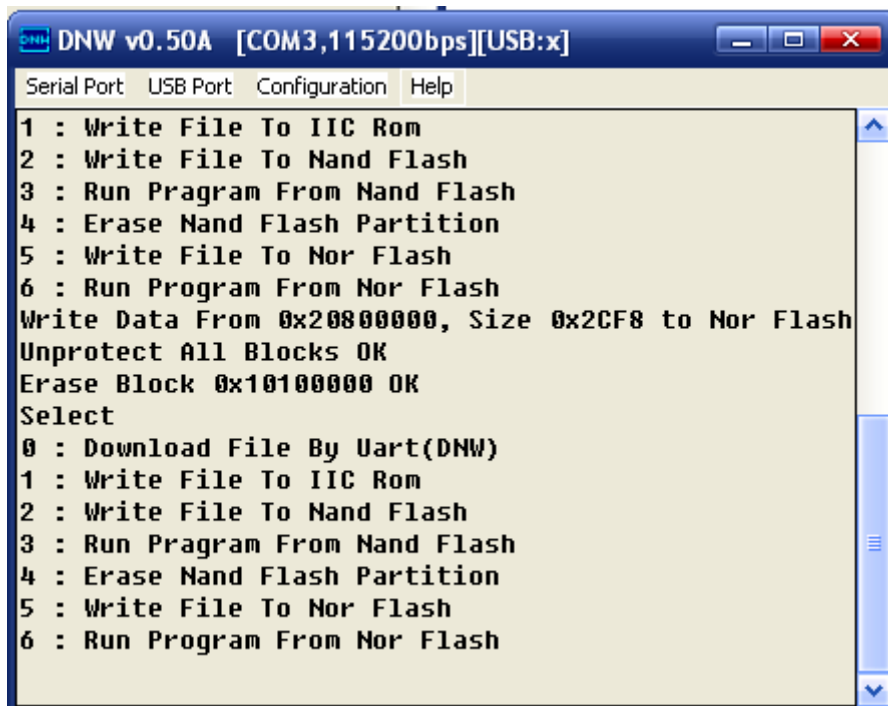
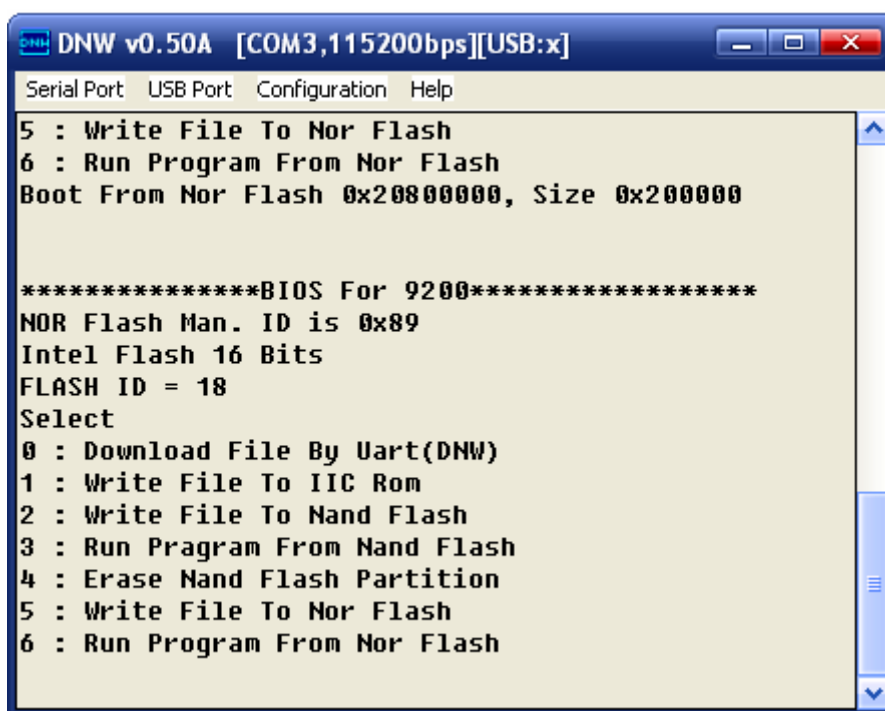


图 20, 选择 6

从图 20 可以看到，bios9200 已经被烧写到 0x1010 0000 地址，即在 Nor FLASH 的地址段，但不是最起始的地址（Nor FLASH 的起始地址是 0x1000 0000）。

下载成功后，我们运行一下，看是否能输出正确信息。
选择 6。

马上出现如图 21 所示的运行结果，和我们程序设计的运行结果一致。说明 bin 文件已经被正确下载并可以正确运行。



```
DNW v0.50A [COM3,115200bps][USB:x]
Serial Port  USB Port  Configuration  Help
5 : Write File To Nor Flash
6 : Run Program From Nor Flash
Boot From Nor Flash 0x20800000, Size 0x200000

*****BIOS For 9200*****
NOR Flash Man. ID is 0x89
Intel Flash 16 Bits
FLASH ID = 18
Select
0 : Download File By Uart(DNW)
1 : Write File To IIC Rom
2 : Write File To Nand Flash
3 : Run Program From Nand Flash
4 : Erase Nand Flash Partition
5 : Write File To Nor Flash
6 : Run Program From Nor Flash
```

图 21，Nor Flash 运行情况

注意：由于 bios9200.bin 是烧写到 NOR FLASH 的 0x1010 0000 这个地址的，而不是 NOR FLASH 的起始地址 0x1000 0000，所以，当 9200 复位后，即便选择从 NOR FLASH 启动，也是不会有输出信息的。因为从 NOR FLASH 启动后，是运行地址 0x1010 0000 的代码，但是实际上该地址是没有任何有效内容的（假设在利用 bios9200 编程 NOR FLASH 前，NOR FLASH 是全空的），这个时候 9200 就取不到有效代码。这点可以通过 JLINK 来读取 FLASH 的内容进行证实，实际读取的内容是从 0x1010 0000 地址处开始，0x1010 0cf0 处结束。

如果要想写入 NOR FLASH 的代码在复位后能从 FLASH 启动，需要更改 bios9200 里面的部分代码，使得烧写到 NOR FLASH 的时候从 0x1000 0000 的地址开始烧写。