

---

# JEDIView for ARM 之 EVB2410 篇

(JEDIView 版本 : Ver1.3)

文档编号 Doc No. :  
修改日期 Data : 2005-9-17  
版本 Rev : 1.2

**MICETEK**

Micetek International Inc.

**Revision History:**

<b>Revision</b>	<b>Date</b>	<b>Comment</b>
<b>1.0</b>	<b>June 18, 2004</b>	<b>Initial Version</b>
<b>1.1</b>	<b>September 30, 2004</b>	<b>Release09302004</b>
<b>1.2</b>	<b>Sep 20,2005</b>	<b>Add EVB2410</b>

---

# 目 录

<b>JEDIView for ARM 开发环境简介</b>	<b>1</b>
如何 XXXXXXXXXXXX .....	1
?? 如何 XXXXXXXXXXXX .....	1
?? 如何 XXXXXXXXXXXX .....	1
<b>第 1 部分: JEDIView for ARM 开发环境介绍</b>	<b>2</b>
<b>第 2 部分: JEDI 开发仿真器硬件介绍</b>	<b>2</b>
1.1 设置 XXX .....	2
1.2 XXXXXX .....	2
1.3 内容 XX .....	3
<b>第 3 部分: 如何在 JEDIView for ARM 中实现高速烧录 FLASH</b>	<b>2</b>
<b>第 4 部分: 如何连接实现超频测试 EVB2410 开发板</b>	<b>2</b>
1.1 准备工作 .....	5

---

## 第一章 JEDIView for ARM 开发环境简介

---

## 第二章 JEDI 开发仿真器硬件说明

### JEDI (JTAG Ethernet Debug Ice)



#### JEDI 简介：

基于ARM的开发目前正成为嵌入式应用领域的一个热点，MICROTEK在前一代产品的基础上，又新推出具有更高性价比的JEDI仿真器。其显著特点是快速应用程序下载，实现快速以太网口与主机通讯，极大方便了远程调试和团队共享的功能。

#### JEDI特点：

##### CPU实时控制

JEDI与ARM通过高性能的JTAG口相连,实现对CPU的控制，不占任何用户资源，高速程序代码下载200-1000KB/S。

##### TCP/IP网络连接

JEDI与主机通讯采用10/100Base-T Ethernet和串口通讯，网口既快速又可实现远程调试。

##### Flash Memory在线编程

JEDI与HITool、JEDIVIEW集成IDE结合，支持多种Flash Memory算法库，并支持用户自定义Flash算法。还可以使用免费的Flash Utility应用程序。

##### 用户寄存器自定义功能

用户可以方便地增加常用的寄存器和周边的寄存器，并可以按位方式显示，对于新CPU更加重要。

##### Target 板自诊断功能

支持对目标Memory Fill, Search, Copy, Test, Compare Data Bus和Address Bus TEST。这些功能对硬件调试非常有用。

##### 硬件断点，支持条件断点和断点计数。

在线Firmware升级，支持最新的ARM或MIPS内核，保护用户投资。

## JEDI开发仿真器套件外观



### 1. 开发套件构成

JEDI 开发仿真器套件包括：

- 1) 一个已测试好的JEDI PROBE 开发仿真器；
- 2) 产品使用说明书一本；
- 3) 一个JTAG 调试头,该调试头可用来烧写和调试程序；
- 4) 一条直连串口线和转接座；
- 5) 一条交叉线网线；
- 8) 一个5V 直流电源；
- 9) 一张JEDI PROBE光盘, 包含有关驱动程序以及使用电子文档；
- 10) 一个包装盒。

### 2. JEDI 开发仿真器资源

硬件技术参数：

- 16/32-bit CPU 内核
- 系统时钟：80MHz

- 一个通道UART/Console 口
- 一个10M以太网网络接口
- JTAG 可工作到40Mhz，自动适应并可以调整工作频率
- 下载速度可达 1M/S (JEDI-II) (300K ~400K JEDIprobe)

## JEDI开发仿真器简介

### 1. 核心板说明

- 1) J1 (旁边写着JTAG) 的20 芯连接器座子是ARM 公司推荐的标准JTAG 接口；
- 2) S1 是复位按键；
- 3) D4 是电源指示灯，核心板正常供电它就会亮；
- 4) D3 通过一个非门连接到CPU 的PWREN 引脚上，用作指示；
- 5) D1 和D2 分别连接到CPU 的GPB5 和GPB6 引脚上，用作系统指示；
- 6) JP1 用作系统启动模式选择。

### 2. 前板说明



- 1) DC output 为电源输出插座，输出直流电压的范围是3~6V (注意由于钽电解电容的耐压值只有16V，所以切记正极、负极的标识，否则仿真器内部将会损坏)；
- 2) Reset 为复位开关，将它按下2~3秒，系统就可以是目标板复位；
- 3) TARGET为标准的JTAG 20PIN接口，通过它仿真器可以对目标板控制或者通讯使用，使用20 芯排线与目标板上面的JTAG口进行一对一连接。用户也可以请根据需要，对照图纸来设置或更改了解排线，默认情况下这些排针都不接短路块；
- 4) Power、Active、COMM 为三个LED指示灯。

### 3. 背板说明



- 1) Power 为电源开关；
- 2) Ethernet 为以太网接口，由于支持有关以太网调试软件连接。
- 3) Rx/Crs、Col/Link 为两个LED指示灯；
- 4) RS232 为串行接口，同标准的RJ15的连接线可以方便的上上位PC机器通讯，也可以按照双工的方式连接调试软件。

#### 4. 左侧板说明



- 1) DC input 为电源输入插入口，内正外负；注意输入直流电压的范围是3~6V，一定不得超过9V，否则将损坏仿真器！另外电源插座的内芯为正极、外壳为负极，极性不符的电源插头也将损坏仿真器。
- 2) WARRANTY 易碎标签是产品的保修凭证，千万注意：个人不要自行打开，一般JEDI PROBE仿真器硬件上作了充分的保护，很不容易损坏；
- 3) Trigger 为硬件触发接口，（特殊用途保留，一般不使用）。

#### 5. 右侧板说明

仿真器右侧方的面板上，暂时无任何标识，以后的版本可能会有新的改动。

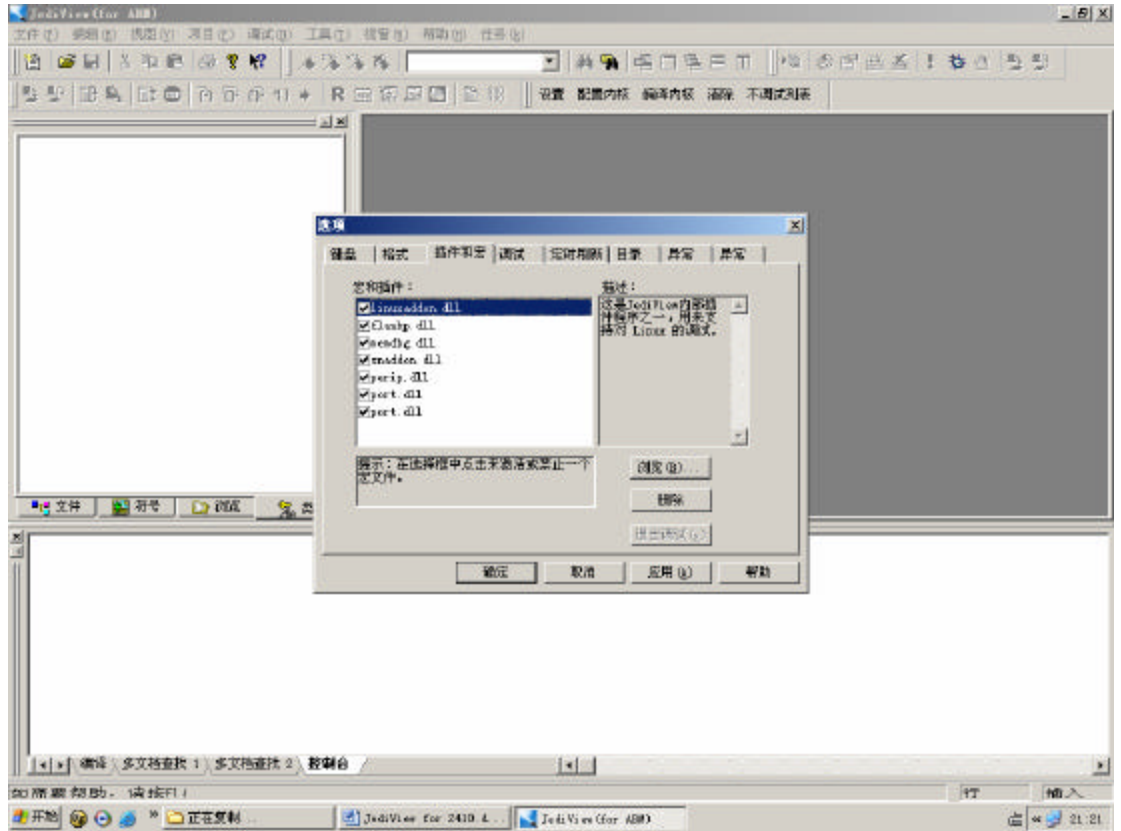
#### 6. 上下板说明



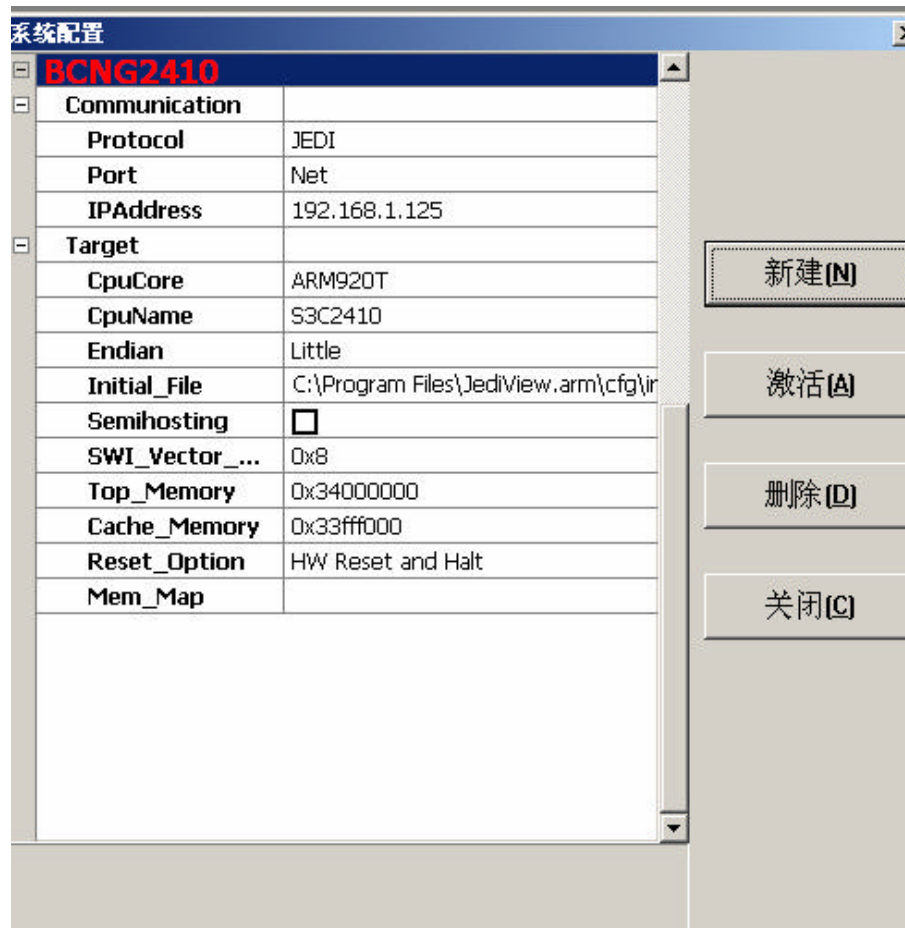
### 第三章 如何在 JEDIVIEW for ARM 中实现高速烧录 FLASH

(待整理完善)

第一步：填加 JEDIVIEW FOR ARM 的 LINUX ADDON 模块：



第二步：设置 JIEDIVIEW FOR ARM 的环境参数：



### 第三步：编写好开发板的初始化脚本：

```

rem *****
rem *           Mdevsoft BCNG2410-I Board Initial File           *
rem *No remap, SDRAM start 0x30000000, ROM start 0x00000000*
rem *           MICETEK (ShenZhen) 2005                         *
rem *****
rem ***SDRAM_Little_16***
!
rem Initial file for S3C2410 Ev Board
long 0x53000000 0x00000000
long 0x4a000008 0xffffffff
long 0x4a00001c 0x000007ff
rem long 0x56000050 0x000055aa
rem long 0x56000058 0x000000ff
rem long 0x4c000014 0x00000003
long 0x4c000000 0x00ffffff
long 0x4c000004 0x00074012
long 0x4c000008 0x00058042

rem memory config Register
long 0x48000000 0x22111120
long 0x48000004 0x00002F50
long 0x48000008 0x00000700
    long 0x4800000c 0x00000700

```

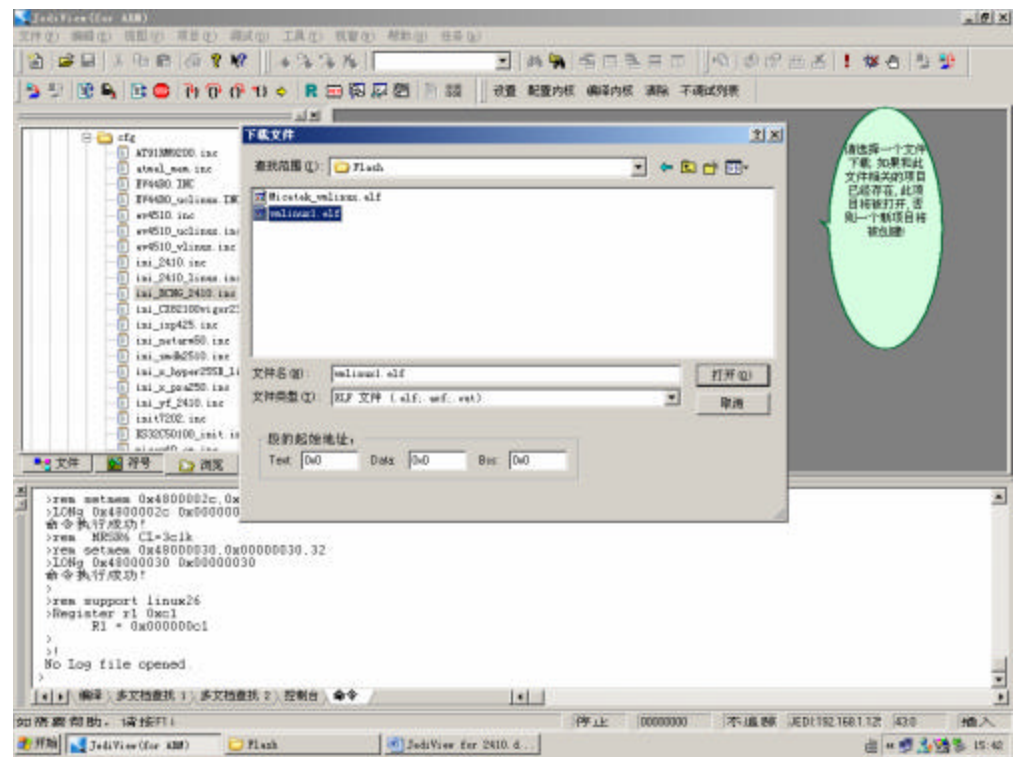
```
long 0x48000010 0x00000700
long 0x48000014 0x00000700
long 0x48000018 0x00000700
long 0x4800001c 0x00018005
long 0x48000020 0x00018005
long 0x48000024 0x008e0459
long 0x48000028 0x00000032
```

```
rem SCLK power saving mode, BANKSIZE 128M/128M, 4-burs;
rem setmem 0x4800002c,0x00000030,32
long 0x4800002c 0x00000030
rem MRSR6 CL=3clk
rem setmem 0x48000030,0x00000030,32
long 0x48000030 0x00000030

rem support linux26
register r1 0xc1
```

!

#### 第四步：下载编译好的内核程序。



#### 第五步：设置下载的地址为：0X30008000



第六步：内核启动后插山 u 盘(不能是 mp3, 因为可能支持不够好, 里面有你要烧的 vivi 等程序)

第七步：执行以下命令(假设 vivi 文件名是 vivi)

```
Mkdir /mnt/uDisk
mount -t vfat /dev/scsi/host0/bus0/target0/lun0/part /mnt/uDisk
eraseall /dev/mtd/0
cp /mnt/uDisk/vivi /dev/mtd/0
eraseall /dev/mtd/1
cp zImage /dev/mtd/1
eraseall /dev/mtd/2
cp root_acs2410.img /dev/mtd/2
然后重新启动就可以了
```

第八步：在超级终端中超作命令行

```
a) enter vivi shell
vivi>
b) use 'param show' command to get current linux kernel parameter
vivi> param show
Number of parameters: 9
name : hex integer
-----
mach_type : 000000c1 193
media_type : 00000003 3
boot_mem_base : 30000000 805306368
baudrate : 0001c200 115200
xmodem_one_nak : 00000000 0
xmodem_initial_timeout : 000493e0 300000
xmodem_timeout : 000f4240 1000000
  ymodem_initial_timeout : 0016e360 1500000
  boot_delay : 01000000 16777216
Linux command line: noinitrd root=/dev/mtdblock/2 console=ttyS0
macaddr=00:0B:51
```

```
:00:0F:FE ip=172.20.9.1::172.20.1.254:255.255.0.0:BCNG2410  
c) use 'param set linux_cmd_line' command to modify linux kernel parameter  
vivi> param set linux_cmd_line "noinitrd root=/dev/mtdblock/2  
console=ttyS0 macaddr=10:0B:51  
:00:0F:FE ip=172.20.9.2::172.20.1.254:255.255.0.0:acs2410_1"
```

NOTE: The double quotation marks("") is necessary.

```
d) use 'param save' command to save parameter table to flash memory.  
vivi> param save  
Found block size = 0x0000c000  
Erasing... .. done  
Writing... .. done  
Written 49152 bytes  
Saved vivi private data
```

root=/dev/mtdblock/2 就是用来指定根文件系统分区的

## 第四章 如何连接实现超频测试 EVB2410 开发板

### 第一步:资料准备.

参看 2410 手册,我们发现控制 CPU 主频的是 MPLLCON 这个寄存器,其有一个详细的参数表,如果将对应的寄存器付上对应的数值的话,我们就可以使得 2410CPU 工作于较高的频率.

It is not easy to find a proper PLL value. So, we recommend referring to the following PLL value recommendation table.

Input Frequency	Output Frequency	MDIV	PDIV	SDIV
12.00MHz	11.289MHz	N/A	N/A	N/A
12.00MHz	16.934MHz	N/A	N/A	N/A
12.00MHz	22.50MHz	N/A	N/A	N/A
12.00MHz	33.75MHz	82 (0x52)	2	3
12.00MHz	45.00MHz	82 (0x52)	1	3
12.00MHz	50.70MHz	161 (0xa1)	3	3
12.00MHz	48.00MHz (note)	120 (0x78)	2	3
12.00MHz	56.25MHz	142 (0x8e)	2	3
12.00MHz	67.50MHz	82 (0x52)	2	2
12.00MHz	79.00MHz	71 (0x47)	1	2
12.00MHz	84.75MHz	105 (0x69)	2	2
12.00MHz	90.00MHz	112 (0x70)	2	2
12.00MHz	101.25MHz	127 (0x7f)	2	2
12.00MHz	113.00MHz	105 (0x69)	1	2
12.00MHz	118.50MHz	150 (0x96)	2	2
12.00MHz	124.00MHz	116 (0x74)	1	2
12.00MHz	135.00MHz	82 (0x52)	2	1
12.00MHz	147.00MHz	90 (0x5a)	2	1
12.00MHz	152.00MHz	68 (0x44)	1	1
12.00MHz	158.00MHz	71 (0x47)	1	1
12.00MHz	170.00MHz	77 (0x4d)	1	1
12.00MHz	180.00MHz	82 (0x52)	1	1
12.00MHz	186.00MHz	85 (0x55)	1	1
12.00MHz	192.00MHz	88 (0x58)	1	1
12.00MHz	202.80MHz	161 (0xa1)	3	1
12.00MHz	266.00MHz	125 (0x7d)	1	1
12.00MHz	268.00MHz	126 (0x7e)	1	1

一般大家是设置成 202.80 MHz,也就是将 MPLLCON 寄存器付上 0xA1031.如果借助 JEDIVIEW 的软件可以在初始化的时候就将频率付上超频的对应的数值.我们这里测试到 266.00MHz,则应该选取 0x7D0101.

因为 Micetek International Inc.公司的 JEDIVIEW FOR ARM 软件,有一个专门的初始化文件供仿真器调用,我们在这里先说明一下这个初始化文件的内容;

```
rem *****
rem * File Name:      GD_2410.inc                **
rem *   JEDIview for ARM on GD2410-I EVB initial file  **
rem *No remap, SDRAM start 0x30008000, ROM start 0x00000000*
rem *   MICETEK (ShenZhen) Office 2005          *
rem *****
rem ***SDRAM_Little_16***
!
```

```
long 0x53000000 0x00000000
```

```
long 0x4a000008 0xffffffff
long 0x4a00001c 0x000003ff
long 0x56000050 0x55aa
long 0x56000058 0x000000ff
long 0x4C000014 0x3
rem LOCKTIME Register
long 0x4c000000 0x00ffffff
rem MPLLCON
rem long 0x4c000004 0xa1031 //原来的数值 为 202MHZ
long 0x4c000004 0x71011 /改为 26MHZ
rem UPLLCON
long 0x4C000008 0x48032
```

```
Register CONTROL 0xc0001078
```

```
rem memory config
```

```
long 0x48000000 0x2211D120
long 0x48000004 0x00000700
long 0x48000008 0x00000700
long 0x4800000c 0x00000700
long 0x48000010 0x00001F7C
long 0x48000014 0x00000700
long 0x48000018 0x00000700
long 0x4800001c 0x00018005
long 0x48000020 0x00018005
long 0x48000024 0x008e0459

long 0x48000028 0x00000032
rem SCLK power saving mode, BANKSIZE 128M/128M, 4-burs;
long 0x4800002c 0x00000030
rem MRSR6 CL=3clk
long 0x48000030 0x00000030
rem MRSR7

long 0x56000000 0x7fffff
long 0x56000010 0x44555
long 0x56000018 0x7ff
long 0x56000020 0xaaaaaaaa
long 0x56000028 0xffff
long 0x56000030 0xaaaaaaaa
long 0x56000038 0xffff
long 0x56000040 0xaaaaaaaa
long 0x56000048 0xffff
long 0x56000050 0x55aa
long 0x56000058 0xff
long 0x56000060 0xff95ffba
```

```
long 0x56000068 0xffff
long 0x56000070 0x2afaaa
long 0x56000078 0x7ff
long 0x56000088 0x22222222
long 0x5600008c 0x22222242
long 0x56000090 0x22222222
```

```
long 0x50000008 0x0
long 0x5000000c 0x0
long 0x50000000 0x3
long 0x50000004 0x245
long 0x50000028 0x1a
```

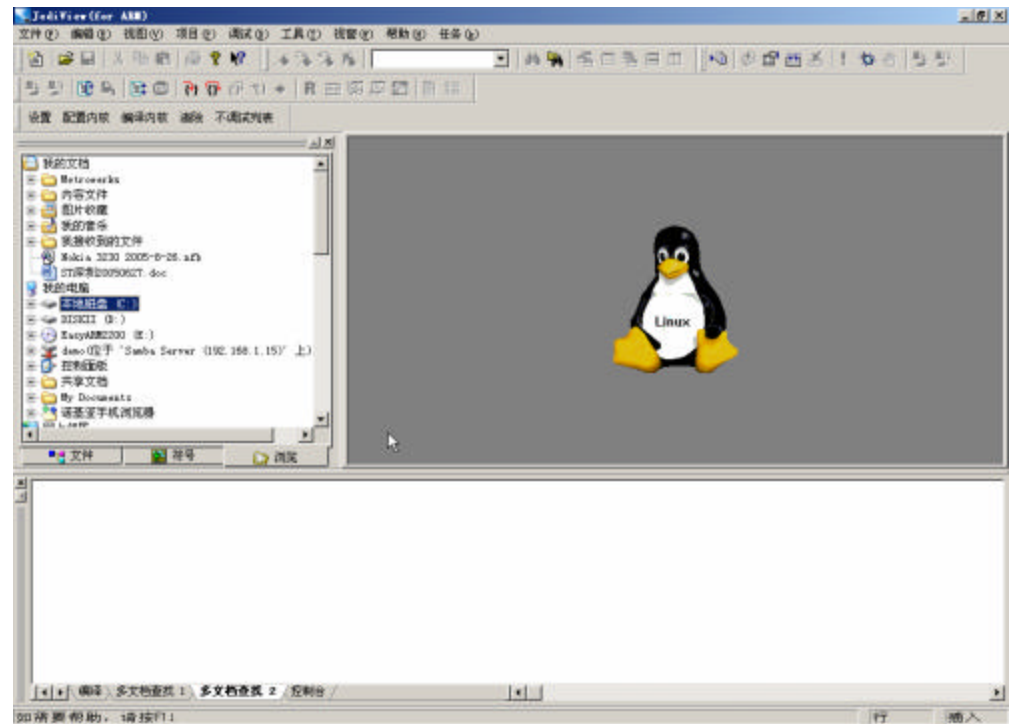
```
rem support linux26
register r1 0xc1
```

!  
我们修改好后,就可以进行下一步了..

## 第二步：准备文件,仿真器以及其他

我们需要一个编译好的带调试信息的 LINUX 内核文件 VMLINUX.ELF ,硬件环境我选择了 JEDI 以及 GD2410 开发板作为硬件环境.

## 第三步：进入 JEDIVIEW FOR ARM 的调试环境.

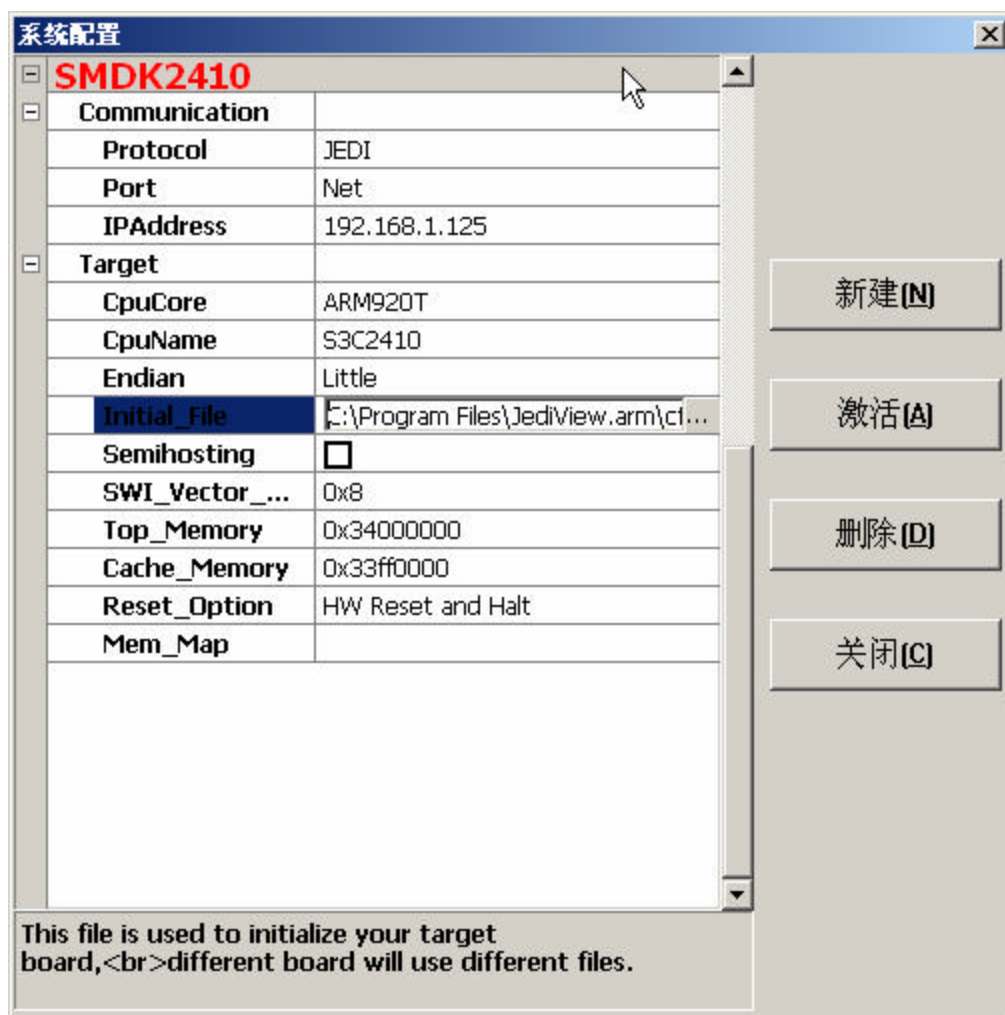


## 配置环境以及设置好参数

现有选择加载的 LINUX 调试模块 LINUX ADDON 这是 MICETEK 独有的调试技术.



还有系统配置窗口,用于设置 IP , CPU , CORE ,INIT 文件.如下图



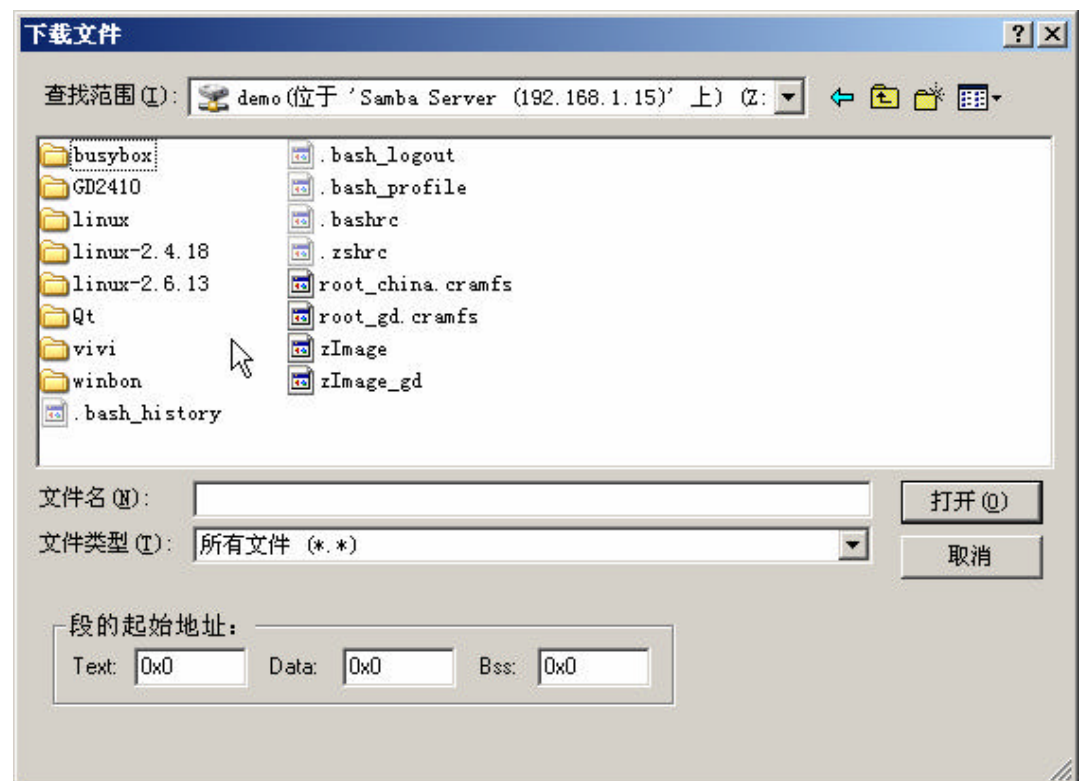


```

(+) Memory
(+) Interrupt
(+) DMA
(-) ClockPower
(+) 04000000 00FFFFFF LOCKTIME PLL lock time count register.
(+) 04000004 0007D011 MPLLCON MPLL configuration register.
(+) 04000008 00048032 UPLLCON UPLL configuration register.
(+) 0400000C 000FFFF0 CLKCON Clock generator control Register.
(+) 04000010 00000004 CLKSLW Slow clock control register.
(+) 04000014 00000003 CLKDIVN Clock divider control register.
(+) LCD
(+) NAND
(+) UART
(+) PWM
(+) USB
(+) WatchDog
(+) IIC
(+) IIS
(+) I/O
(+) RTC
(+) A/D
(+) SPI
(+) SDI

```

然后,我们下载一个带调试信息的 VMLINUX 文件.到开发板上 SDRAM 的 0X30008000 地址.



名称	大小	类型	修改日期
ldscript_rom.ld	9 KB	LD 文件	2003-12-26 18:09
MAINTAINERS	58 KB	文件	2005-8-29 7:41
Makefile	44 KB	文件	2005-9-8 17:28
README	15 KB	文件	2005-8-29 7:41
REPORTING-BUGS	3 KB	文件	2005-8-29 7:41
System.map	448 KB	MAP 文件	2005-9-11 17:38
<b>vmlinux</b>	12,151 KB	文件	2005-9-11 17:38
vmlinux.mpj	4 KB	JediView Project	2005-9-11 22:58
vmlinux.mws	1 KB	JediView 工作空间	2005-9-14 20:31

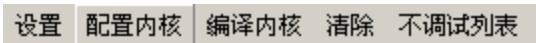
下载完成后,窗口的光标会指向并显示 30008000 地址出的程序,此刻还是汇编代码.

```

➔ 30008000 E321F0D3 msr      cpsr_c,#0xd3
30008004 EB000051 bl        0x30008150
30008008 E1B0A005 movs     r10,r5
3000800C 0A00004D beq      0x30008148
30008010 EB000066 bl        0x300081b0
30008014 E1B08005 movs     r8,r5
30008018 0A00004A beq      0x30008148
3000801C EB00002D bl        0x300080d8
30008020 E59FD004 ldr      r13,#0x3000802c
30008024 E28FE068 add      r14,pc,#0x68
30008028 E28AF00C add      pc,r10,#0xc
3000802C C0008050 andgt   r8,r0,r0,asr r0
30008030 C01F4000 andgts  r4,pc,r0
30008034 C01F4000 andgts  r4,pc,r0
30008038 C0254EC0 eorgt   r4,r5,r0,asr #0x1d
3000803C C026DB48 eorgt   r13,r6,r8,asr #0x16
30008040 C025692C eorgt   r6,r5,r12,lsr #0x12
30008044 C02568FC strgtd  r6,[r5],-r12
30008048 C01F6F9C mulgts  pc,r12,pc
3000804C C01F5FF8 ldrgtsh r5,[pc],-r8
30008050 E24F3028 sub      r3,pc,#0x28
30008054 E8B300F0 ldmia   r3!,{r4-r7}
30008058 E1540005 cmp      r4,r5
3000805C 11550006 cmpne   r5,r6

```

如果需要参看源程序代码,需要再作下列设置:  
 点击下面的工具条的[设置]选项



将源文件路径填写正确即可.



此时如果将下面工具条上的全速运行按钮点击,就可以让下载的程序无中断的运行了.



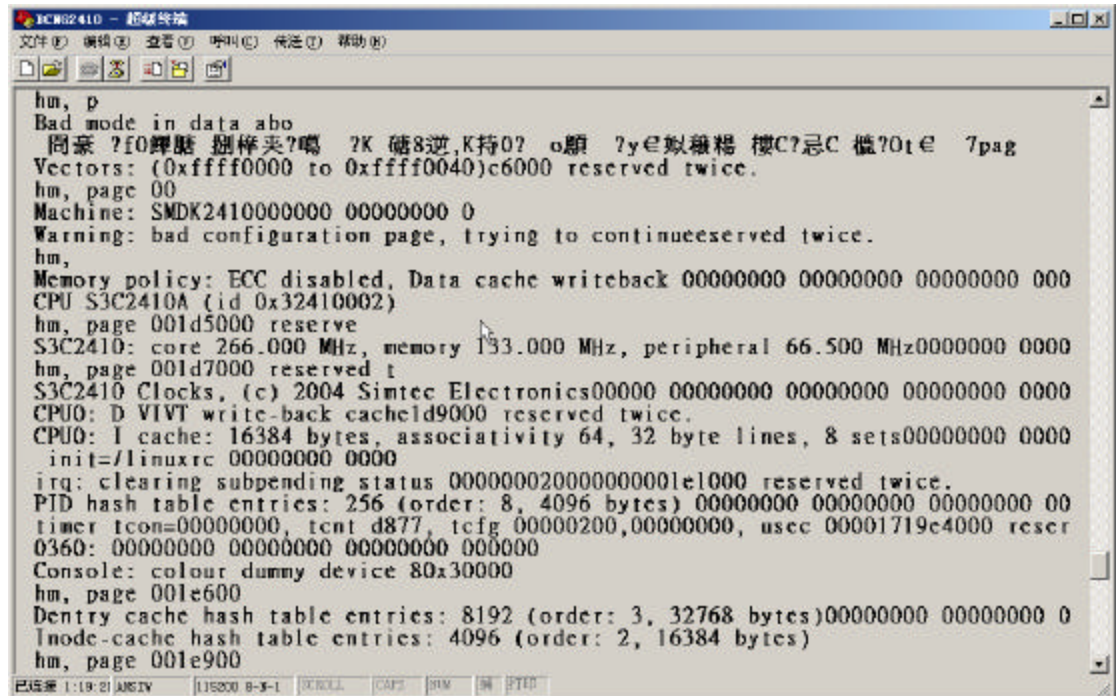
我们可以在预先打开的超级终端程序中,看到串口显示的 CPU 运行状态了..  
 这里我们可以捕捉到如下信息,供大家参考:

```

Linux version 2.6.13 (demo@localhost.localdomain) (gcc version 3.3.2) #89 Sun Sep
11 17:37:54 CST 2005
CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)
Machine: SMDK2410
Warning: bad configuration page, trying to continue
Memory policy: ECC disabled, Data cache writeback
CPU S3C2410A (id 0x32410002)
S3C2410: core 266.000 MHz, memory 133.000 MHz, peripheral 66.500 MHz

```

S3C2410 Clocks, (c) 2004 Simtec Electronics  
 CPU0: D VIVT write-back cache  
 CPU0: I cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets  
 CPU0: D cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets  
 Built 1 zonelists  
 Kernel command line: noinitrd root=/dev/mtdblock3 console=ttySAC0,115200  
 mem=63M init=/linuxrc  
 irq: clearing subpending status 00000002  
 PID hash table entries: 256 (order: 8, 4096 bytes)  
 timer tcon=00000000, tcnt d877, tcfg 00000200,00000000, usec 00001719  
 Console: colour dummy device 80x30  
 Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)  
 Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)  
 Memory: 63MB = 63MB total  
 Memory: 61440KB available (1880K code, 486K data, 84K init)  
 Mount-cache hash table entries: 512  
 CPU: Testing write buffer coherency: ok  
 Unhandled fault: alignment exception (0x003) at 0xe353043e  
 Internal error: : 3 [#1]  
 Modules linked in:  
 CPU: 0.....  
 .....  
 .....



最高我们经过测试,发现工作到 270MHZ 的时候,外围器件无法正常工作了..

```
S3C2410 - 超級終端
文件(F) 編輯(E) 查看(V) 呼叫(C) 代理(A) 幫助(H)
[Icons]
敲<K
  譯擊樓.€ 鑽€? 9歲 祭悍q? L7祭b? 0t;f鯁linux version 2.6.13 (demo@localhos
t.localdomain) (gcc version 3.3.2) #89 Sun Sep 11 17:37:54 CST 2005
CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)
Machine: SMDK2410
Warning: bad configuration page, trying to continue
Memory policy: ECC disabled, Data cache writeback
CPU S3C2410A (id 0x32410002)
S3C2410: core 270.000 MHz, memory 135.000 MHz, peripheral 67.500 MHz
S3C2410 Clocks, (c) 2004 Sintec Electronics
CPU0: D VIVT write-back cache
CPU0: I cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
CPU0: D cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
Built 1 zonelists
Kernel command line: noinitrd root=/dev/mtdblock3 console=ttySAC0,115200 mem=63M
init=/linuxrc
irq: clearing subpending status 00000002
PID hash table entries: 256 (order: 8, 4096 bytes)
timer tcon=00000000, tent dbb9, tcfg 00000200,00000000, uscc 000016c1
Console: colour dummy device 80x30
Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Unhandled fault: alignment exception (0x003) at 0xa0000c5
-
已连接 1:30:11 AMETV | 115200 8-N-1 | [SCROLL] [CAPS] [NUM] [MENU] [PTED]
```