



# OBT-BSP-VxWorks5.4 应用开发手册

V 2.0

珠海欧比特控制工程股份有限公司

2010年10月17日

---

# 目 录

<b>1 引言</b> .....	<b>1</b>
1.1 编写目的.....	1
1.2 定义.....	1
<b>2 用途</b> .....	<b>1</b>
<b>3 软件安装</b> .....	<b>2</b>
<b>4 创建ROM映像</b> .....	<b>3</b>
4.1 新建工程.....	3
4.2 选择BSP.....	5
4.3 修改VxWORKS的宏.....	5
4.4 设置波特率.....	6
4.5 添加应用程序.....	7
4.6 设置应用程序入口.....	8
4.7 选择映像类型.....	9
4.8 创建映像文件.....	10
<b>5 映像烧写及运行</b> .....	<b>11</b>
5.1 烧写最终映像方法.....	11
5.2 启动VXWORKS.....	14
<b>6 移植注意事项</b> .....	<b>14</b>
6.1 不同系统时钟频率的修改.....	14
6.2 不同SRAM数据线宽度的修改.....	15
6.3 不同SRAM大小的修改.....	15

# 1 引言

## 1.1 编写目的

这份手册向读者介绍如何使用 OBT-BSP-VxWorks5.4，主要介绍在 Tornado 集成开发环境中如何开发针对 S698 系列开发板的 VxWorks 5.4 应用软件, 以及生成最终烧写程序的过程。

读者应当具备 C/C++ 编程基础, 并且了解软件开发过程中的编译, 连接, 调试等概念。

## 1.2 定义

S698: 由珠海欧比特软件工程有限公司设计的基于 SPARC V8 指令集的嵌入式处理器系列, 有 S698-ECR、S698-MIL、S698P4、S698PM 等多种型号。

DUSMON :SPARC 架构 CPU 的调试软件, 能在线调试或下载程序。

# 2 用途

OBT-BSP-VxWorks5.4是针对S698系列单核处理器的VxWorks 5.4操作系统的板级支持包 (BSP), 其具体功能包括:

- (1) 目标板硬件初始化, 主要是CPU 的初始化, 为整个系统提供底层硬件支持;
- (2) 为操作系统提供设备驱动程序和系统中断服务程序;
- (3) 定制操作系统的功能, 为软件系统提供一个实时多任务的运行环境;
- (4) 初始化操作系统, 为操作系统的正常运行做好准备。

OBT-BSP-VxWorks5.4 适用的处理器包括: S698M、S698-ECR、S698-MIL、S698-T 等, OBT-BSP-VxWorks5.4 基于的内核版本为 VxWorks 5.4, 集成开发环境为 Tornado2.0。

集成开发环境 Tornado2.0 用于 VxWorks 嵌入式系统的交叉开发，Tornado 使用了 Target Server-Agent 模式来建立主机和目标机的交叉开发环境，同时它也是一个开放的可扩展环境，很容易集成第三方的开发工具，用户也可以按自己的习惯定制开发环境。

Tornado2.0 提供嵌入式软件开发过程中的工程项目管理、源程序编辑、编译、链接、运行、调试等功能。

新版本的 Tornado 集成开发环境还可以加入许多图形化的第三方开发工具，其中包括：

1. 调试器 Cross Wind;
2. 强有力的命令行执行方式 WindShell;
3. 系统对象检查工具 Browser;
4. 模拟器 Simulator;
5. 软件逻辑分析仪 WindView;
6. 源代码分析仪 (WindNavigator)。

### 3 软件安装

1. **安装开发环境：**首先安装 For SPARC 的 Tornado 软件，安装包在 TDK-12841-02 目录中，点击 Setup.exe 开始安装，建议安装路径为 D:\Tornadosparc，其他步骤按照默认设置。



图 3-1 Tornado2.0 的安装

2. **安装补丁程序:** 点击 vxwin2kpathch 目录下的 Setup.exe 文件安装补丁。
3. **安装 BSP 包:** 将光盘 BSP 目录中 target 目录整个拷贝并覆盖安装后 Tornado 目录下的 target 目录。
4. **拷贝工具软件:** 需要将 Leon BSP 包中的十六进制转换软件 hex 拷贝到系统目录下 (既将 D:\Tornadosparc\target\config\leon\bin 里面的 hex.exe 文件复制到 D:\Tornadosparc\host\x86-win32\sparc-wrs-VxWorks\bin)。

## 4 创建ROM映像

### 4.1 新建工程

软件安装完成后, 点击桌面图标打开 Tornado 开发环境。

选择创建一个 VxWorks 映像工程 (Create a bootable VxWorks image), 然后点 “ok”。如下图所示:

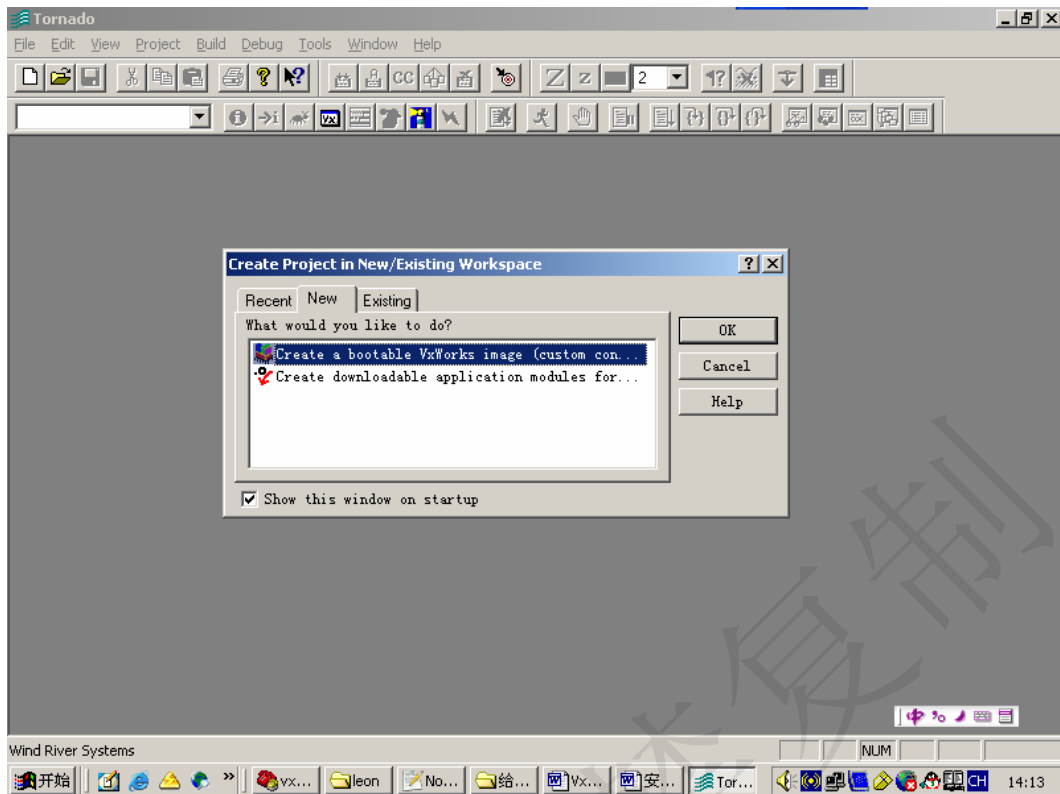


图 4-1 Tornado2.0 的安装

接下来，在弹出的窗口中输入工程名称（默认为 Project1）。如下图所示：

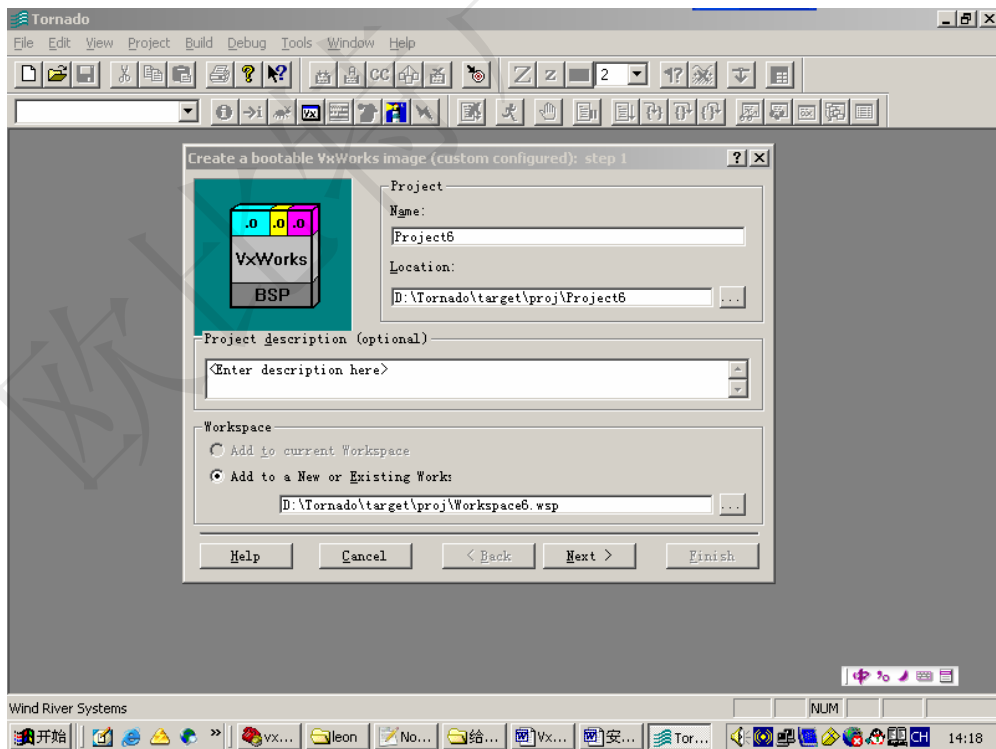


图 4-2 输入工程名

## 4.2 选择BSP

在接下来的弹出窗口中选择工程 BSP 类型：**leon**。如下图所示：

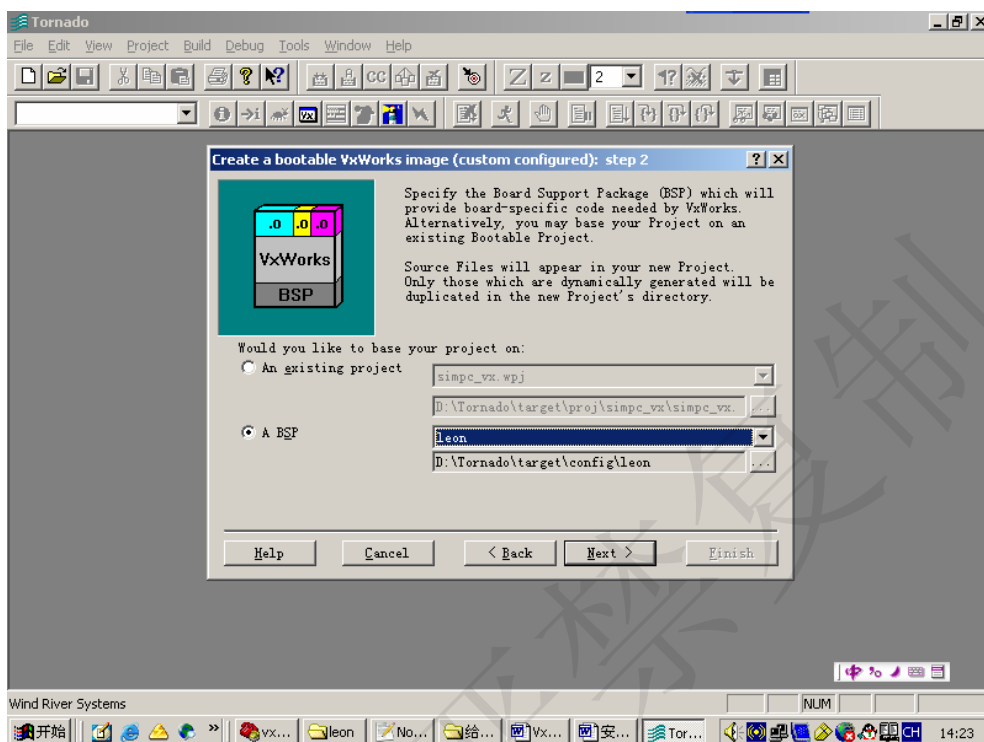


图 4-3 选择 BSP 类型

选择后点“Next”，再在弹出的窗口中，点击“Finish”。这样便完成了一个基于 leon BSP 的 VxWorks 应用工程的创建。

## 4.3 修改VxWorks的宏

在 Tornado 的工作区中，分有 Files、VxWorks 和 Builds 三个页。点击 Builds 页，然后右击“default”打开 Build 的属性对话框。在属性对话框中，选择 Macro 页，修改以下宏：

**ADDED\_CFLAGS:** 改为 `-g -O0 -DDEBUG -Wall -DRAM_DATA_ADRS = 0x40010000`。

**RAM\_HIGH\_ADRS:** 改为 `40080000`（`40080000` 中 `80000` 这个值是实际 RAM 的一半，开发板 RAM 的大小位 `0x100000`）。

**RAM\_LOW\_ADRS:** 改为 `40004000`。

## 4.4 设置波特率

1. 在 Workspace 工作区选择 VxWorks 配置页，选中 WDB serial connection 组件，点击鼠标右键，选择 Properties 页，在弹出的窗口中设置波特率为 57600，（注：VxWorks 的 Serial 组件中设置波特率为 57600，否则 VxWorks 的 Logo 不能打印输出）。如下图所示：

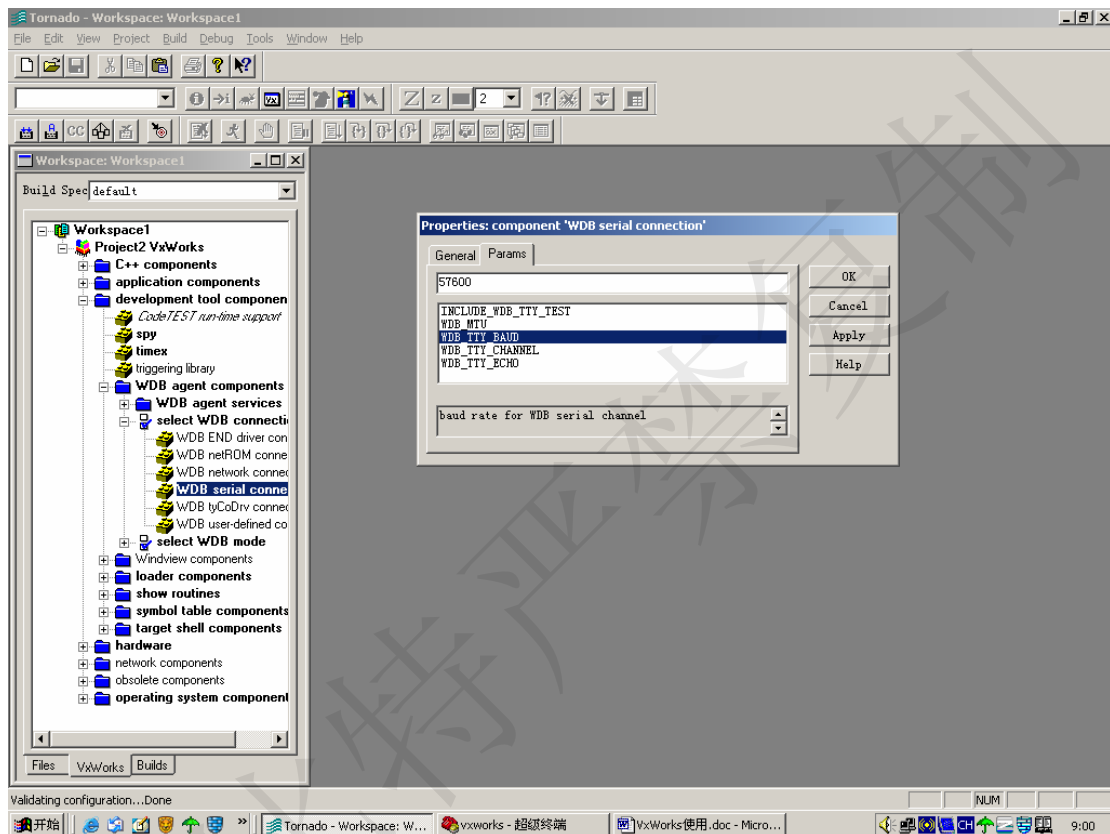


图 4-4 设置通讯波特率 1

2. 在 VxWorks 工作区中的 hardware 组件里面选择 SI0，再设置波特率为 57600。如下图所示：

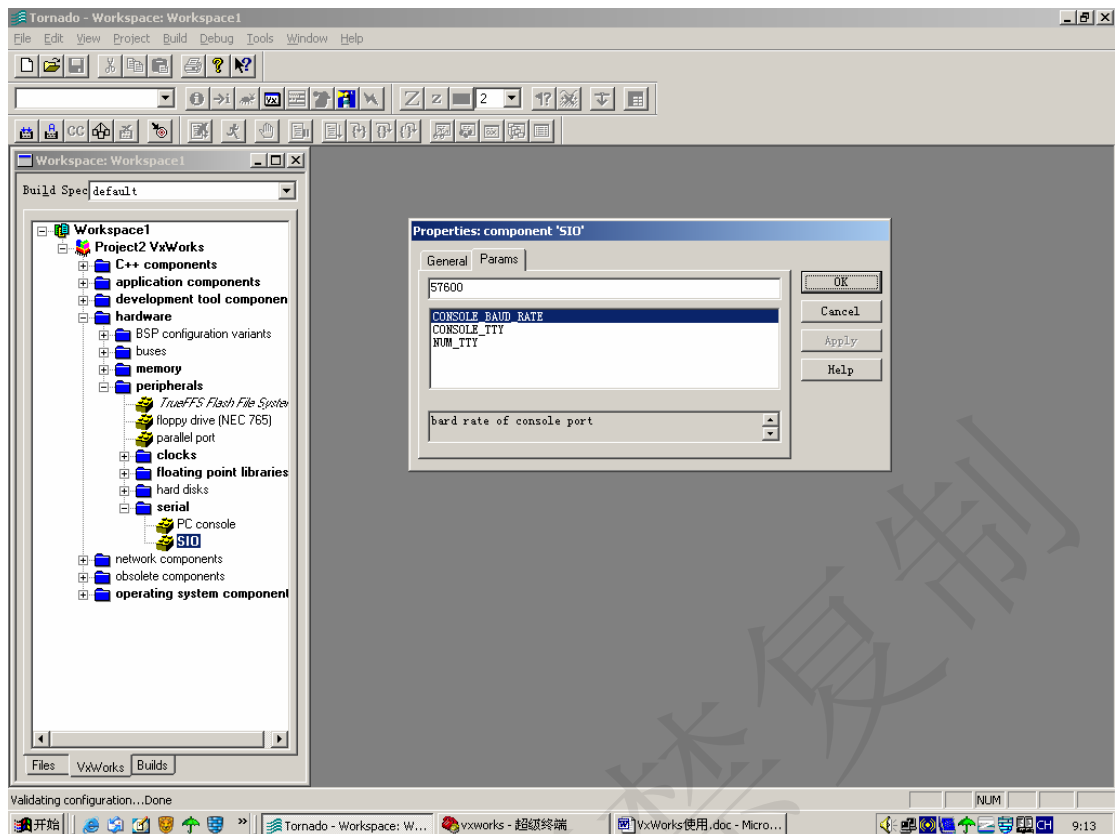


图 4-5 设置通讯波特率 2

## 4.5 添加应用程序

工程新建后，系统自动生成一些源文件，用户可以向工程中添加自己编写的应用程序源文件，一起建立 VxWorks 映像。鼠标点中工程，右键点击“add Files”菜单项，通过文件对话框选择文件的所在目录，就可以添加源文件了。如下图所示：

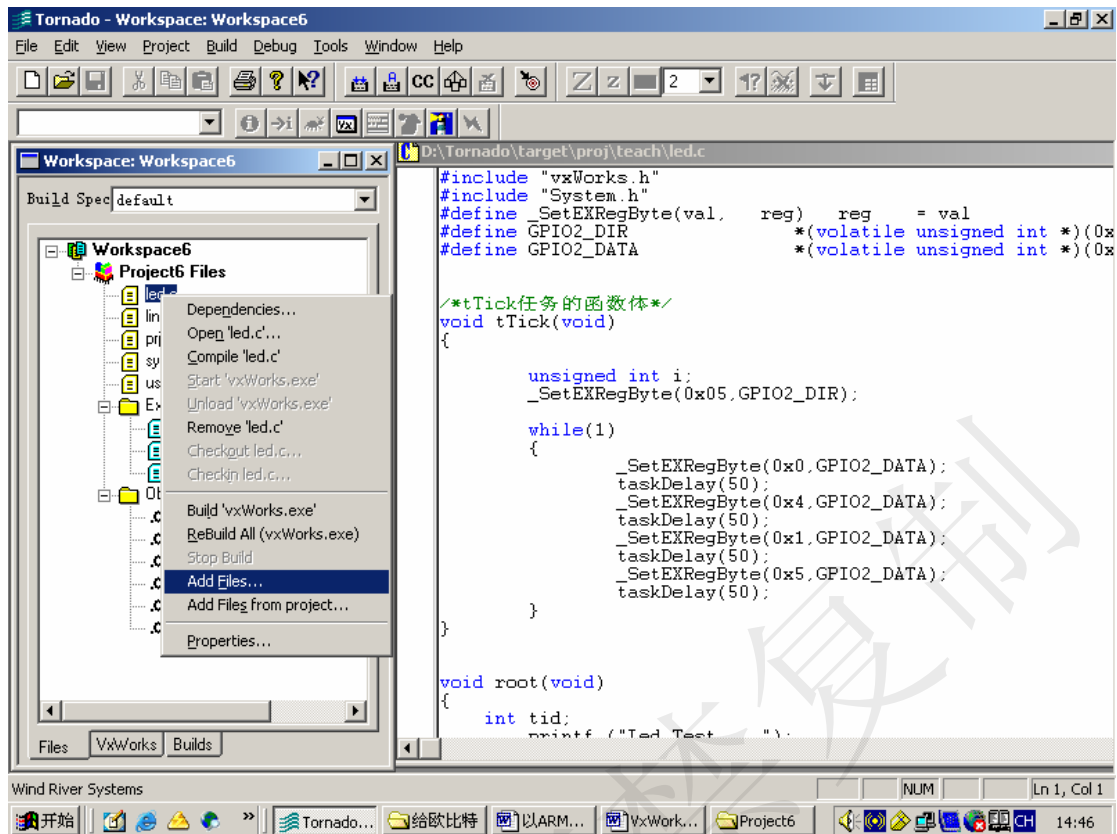


图 4-6 向工程中添加应用源文件

## 4.6 设置应用程序入口

添加完应用源文件后,还需要在 `usrAppInit.c` 的 `usrAppInit` 函数中添加应用入口函数,以便运行时 VxWorks 操作系统加载完成后能够自动跳转到用户应用程序中。如下图所示:

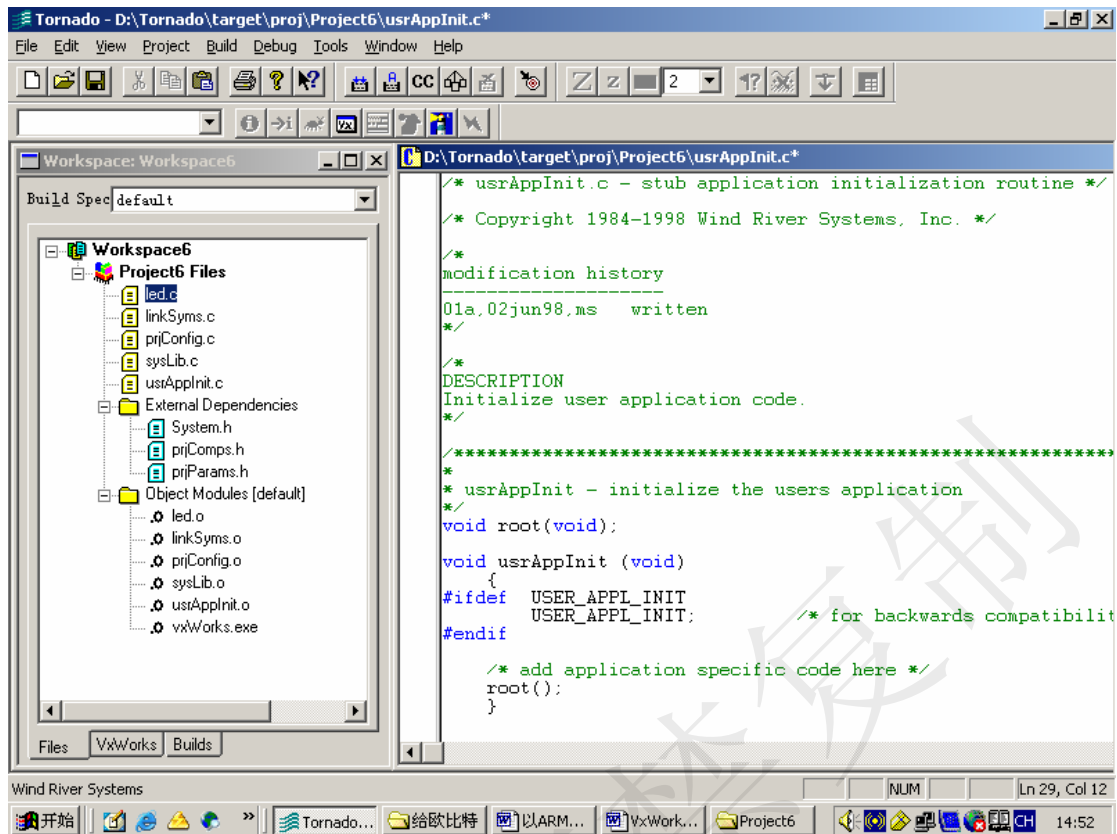


图 4-7 添加应用入口

上图中“root()”为用户应用程序中的第一个入口程序。当然在 usrAppInit() 之前应加上必要的 include 头文件，才能在后面的编译中通过。

## 4.7 选择映像类型

映像编译之前需要选择映像类型，通常情况下都需要选择 VxWorks\_romCompress 类型（既可以在 rom 中直接启动的压缩的映像）。

首先在 VxWorks 工作区中点击“Builds”页，然后双击“default”子项，在弹出的窗口选择 rules 设置页，Rule 选择框选择 VxWorks\_romCompress 项，然后点击 OK。如下图所示：

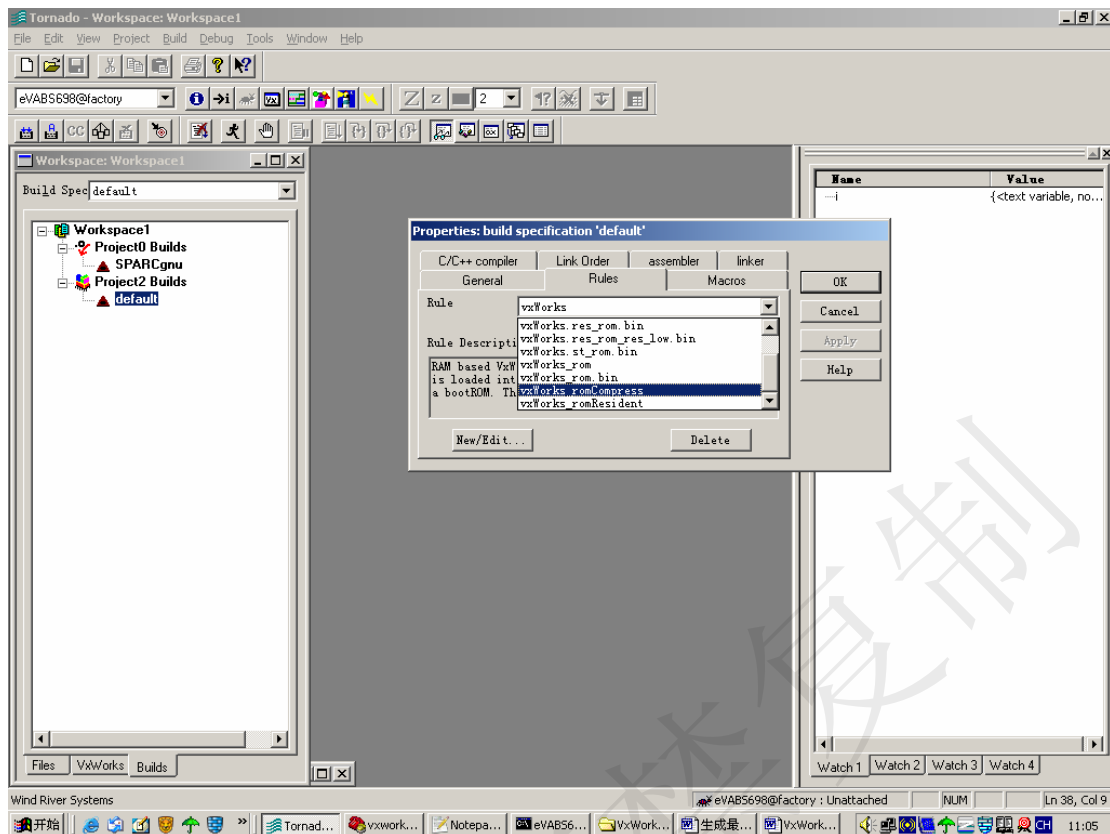


图 4-8 选择映像类型

## 4.8 创建映像文件

选择 Build/Build 菜单，创建下载用的 VxWorks\_romCompress 映像，系统开始编译映像文件，VxWorks\_romCompress 映像文件生成后将默认保存在工程目录下的 default 目录中。如下图所示：

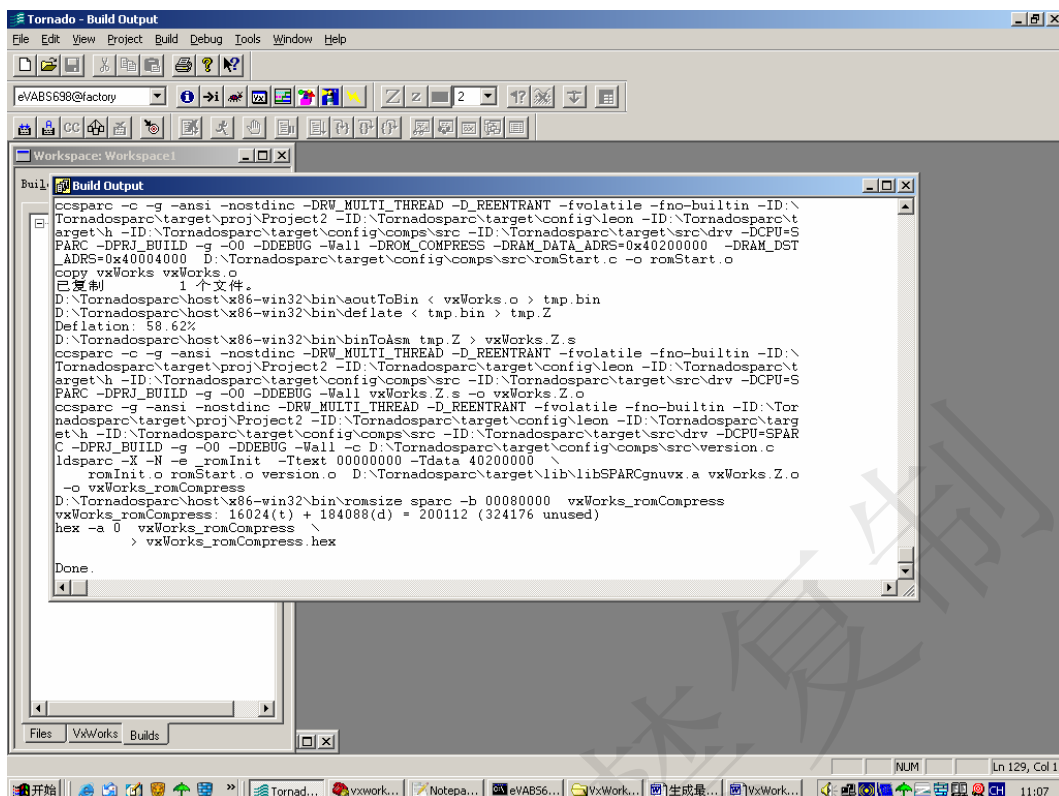


图 4-9 创建 VxWorks 映像

## 5 映像烧写及运行

### 5.1 烧写最终映像方法

用户可以通过烧写器，将生成的 VxWorks\_romCompress\_映像烧写到 ROM 中，或通过 DSU 调试口下载映像文件。

通过 DSU 调试口烧写程序步骤如下：

1. 生成映像文件：首先通过上文讲述方法生成 VxWorks 映像文件 VxWorks\_romCompress.hex。

2. 生成 BIN 文件：使用 OBJCOPY 命令将 VxWorks\_romCompress.hex 转换成 VxWorks\_romCompress.bin 在 CYGWIN 环境中的文件目录下输入：  
`sparc-rtems-objcopy -O binary VxWorks_romCompress.hex VxWorks_romCompress.bin`

3. 生成烧写文件：通过 mkflash 命令根据不同 flash 使用不同命令参数生成烧写文件 flash.out。

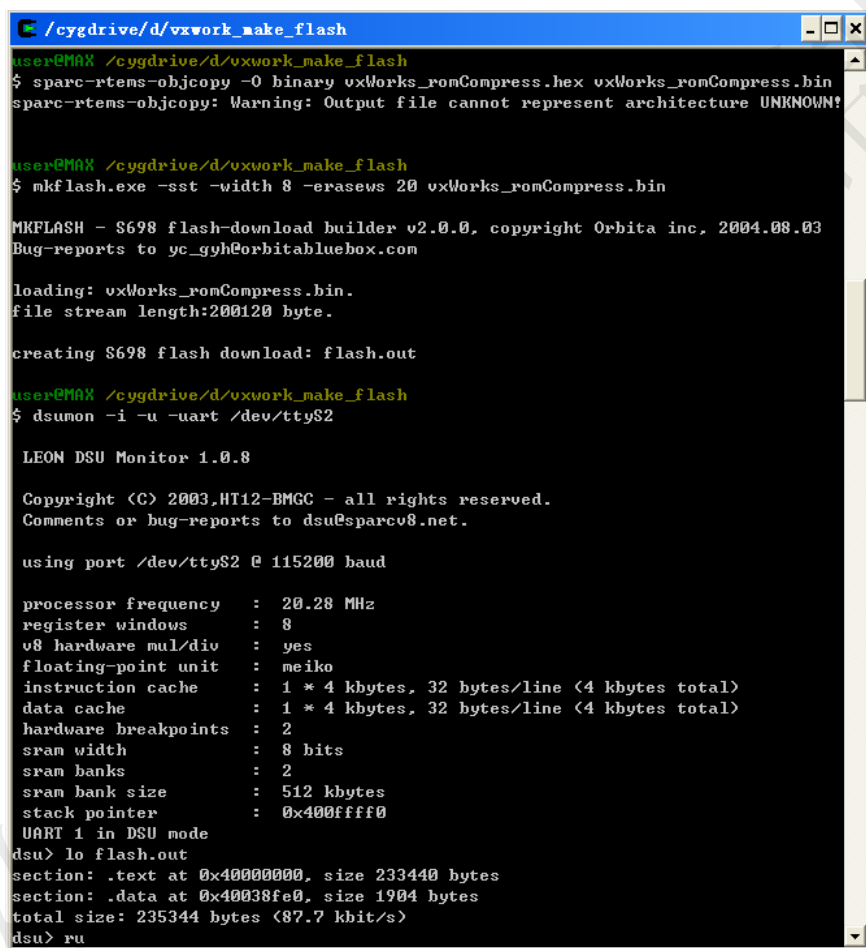
SST 公司 8 位 FLASH 生成烧写文件命令:

```
mkflash.exe -sst -width 8 -erasews 20 VxWorks_romCompress.bin
```

AMD 公司 16 位 FLASH 生成烧写文件命令:

```
mkflash.exe -amd -width 16 -erasews 100 vxWorks_romCompress.bin
```

4. **完成烧写操作:** 使用 V8MON 或 DSUMON 将 flash.out 程序下载到内存后再运行, 完成对 flash 的烧写。整个烧写的过程如下图所示:



```
user@MAX /cygdrive/d/vxwork_make_flash
$ sparc-rtens-objcopy -O binary vxWorks_romCompress.hex vxWorks_romCompress.bin
sparc-rtens-objcopy: Warning: Output file cannot represent architecture UNKNOWN!

user@MAX /cygdrive/d/vxwork_make_flash
$ mkflash.exe -sst -width 8 -erasews 20 vxWorks_romCompress.bin

MKFLASH - S698 flash-download builder v2.0.0, copyright Orbita inc, 2004.08.03
Bug-reports to yc_gyh@orbitabluebox.com

loading: vxWorks_romCompress.bin.
file stream length:200120 byte.

creating S698 flash download: flash.out

user@MAX /cygdrive/d/vxwork_make_flash
$ dsumon -i -u -uart /dev/ttyS2

LEON DSU Monitor 1.0.8

Copyright (C) 2003,HI12-BMGC - all rights reserved.
Comments or bug-reports to dsu@sparcv8.net.

using port /dev/ttyS2 @ 115200 baud

processor frequency : 20.28 MHz
register windows : 8
v8 hardware mul/div : yes
floating-point unit : meiko
instruction cache : 1 * 4 kbytes, 32 bytes/line (4 kbytes total)
data cache : 1 * 4 kbytes, 32 bytes/line (4 kbytes total)
hardware breakpoints : 2
sram width : 8 bits
sram banks : 2
sram bank size : 512 kbytes
stack pointer : 0x400ffff0
UART 1 in DSU mode
dsu> lo flash.out
section: .text at 0x40000000, size 233440 bytes
section: .data at 0x40038fe0, size 1904 bytes
total size: 235344 bytes (87.7 kbit/s)
dsu> ru
```

图 5-1 通过 dsumon 烧写 VxWorks 映像 1

```

/cygdrive/d/vxwork_make_flash
register windows      : 8
v8 hardware mul/div  : yes
floating-point unit  : meiko
instruction cache     : 1 * 4 kbytes, 32 bytes/line (4 kbytes total)
data cache           : 1 * 4 kbytes, 32 bytes/line (4 kbytes total)
hardware breakpoints : 2
sram width           : 8 bits
sram banks           : 2
sram bank size       : 512 kbytes
stack pointer        : 0x400ffff0
UART 1 in DSU mode
dsu> lo flash.out
section: .text at 0x40000000, size 233440 bytes
section: .data at 0x40038fe0, size 1904 bytes
total size: 235344 bytes (87.7 kbit/s)
dsu> ru
erasing the flash,Please wait.
erase finished.

begin write to flash,wait.
.....
end write to flash.
write total 200120 bytes data to flash.

Program exited normally.
dsu>
    
```

图 5-2 通过 dsumon 烧写 VxWorks 映像 2

5. 特殊情况处理：在某些情况下可能应为 DSU 自动初始化 CPU 不能符合硬件配置导致烧写失败。用户可以在 `lo flash.out` 命令前输入入命令修改下以下几个参数，保证烧写能正常。例如 32 位 RAM,16 位 FLASH 的硬件可按图中修改：

```

dsu> wmem 0x80000000 0x9ff
dsu> mcfg1 0x9ff
    mcfg1 = 0x000009ff
dsu> wmem 0x80000004 0x9060
dsu> mcfg2 0x9060
    mcfg2 = 0x00009060
dsu> stack 0x401ffff0
    stack pointer: 0x401ffff0
    
```

图 5-3 在 32 位 RAM,16 位 FLASH 情况下配置

## 5.2 启动VXWORKS

烧写完成后将电脑串口和开发板的串口 1 相连。打开电脑串口调试助手，设置连接串口波特率 56000，8 位数据位，1 位停止位，无奇偶效验。重新启动开发板，如果编译和烧写正常，串口将出现大量字符信息输出。如下图所示：

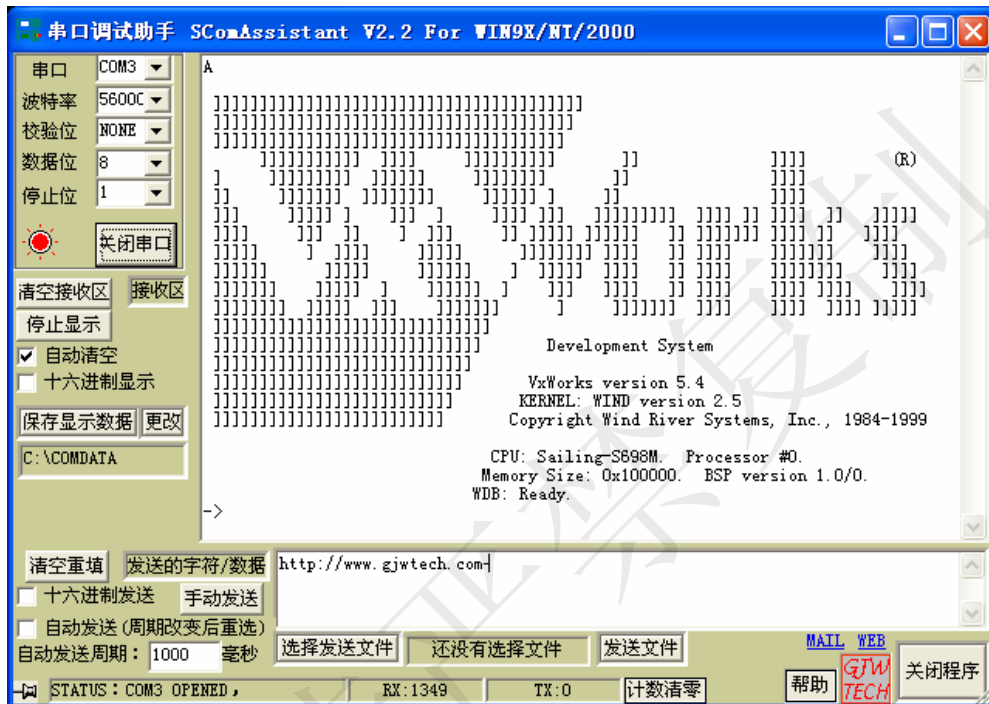


图 5-4 VxWorks 映像运行输出信息

## 6 移植注意事项

不同的开发板可能涉及到一些 BSP 包的移植，针对 S698 系列开发板的移植主要涉及到部分文件和部分参数，参考下文。

### 6.1 不同系统时钟频率的修改

1. 根据实际值设置修改 eVAB695.h 中的 SYS\_CLK\_FREQ 参数（下图黄色部分）：

```
#define BBRAM          ((volatile char *) 0xfffb0000)

#define SYS_CLK_FREQ  20000000 /* CLK2 = 20MHz */
#define WD_CLK_FREQ   3686400 /* */
```

2. 在 romInit.s 中修改串口打印波特率换算值（根据晶振频率计算出串口设置寄存器的值）：

```
set 43,%13 /*Baudrate 57600 */
```

## 6.2 不同SRAM数据线宽度的修改

修改 romInit.s 中的 MCR2 值（下图黄色部分，MCR2 值要根据数据线位宽进行换算）：

```
st %15,[%14] /*configure MCR1*/
//gong set 0x000106f,%15
set 0x00000c4f,%15
st %15,[%14+4] /*configure MCR2*/
```

## 6.3 不同SRAM大小的修改

1. 根据目标板 SRAM 大小修改 config.h 中的 RAM\_HIGH\_ADRS 参数值（黄色部分），RAM\_HIGH\_ADRS 一般可设为 LOCAL\_MEM\_SIZE 大小的一半。

```

#undef LOCAL_MEM_AUTOSIZE          /* run-time memory sizing */

#define LOCAL_MEM_SIZE              0x00100000 /* 1M memory */

.....

#if (LOCAL_MEM_SIZE == 0x00200000)
    #define RAM_HIGH_ADRS          0x40100000 /* RAM address for bootrom
*/
#else
    #define RAM_HIGH_ADRS          0x40080000 /* RAM address for bootrom

```

2. 根据目标板 SRAM 大小修改 config\comps\VxWorks\00bsp.cdf 中的黄色部分，该部分是设置工程的内存大小。

```

Parameter LOCAL_MEM_SIZE {
    NAME          local memory size
    SYNOPSIS      Fixed (static) memory size
    DEFAULT       0x00100000
}

```

3. 修改工程中的 RAM\_HIGH\_ADRS 参数（与 config.h 中的保持一致）。如下图所示：

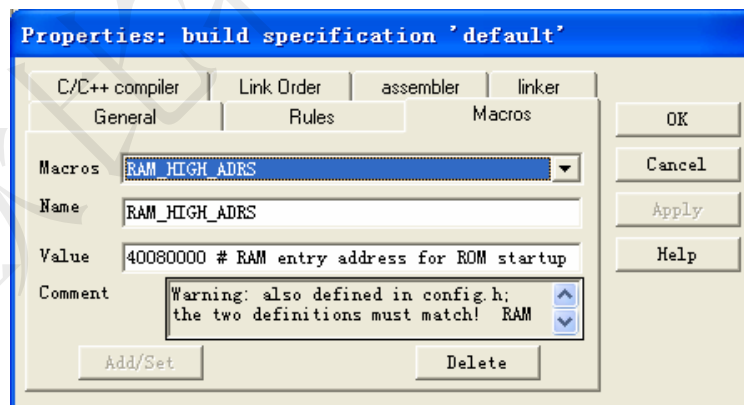


图 6-1 修改 RAM\_HIGH\_ADRS 参数