

# AT91RM9200 BSP Programmer Guide

## NOTICE

© 2005 Coordinate Co., Ltd. All intellectual property rights in this publication are owned by Coordinate Co., Ltd. and are protected by People's Republic of China copyright laws, other applicable copyright laws and international treaty provisions. Coordinate Co., Ltd. retains all rights not expressly granted.

No part of this publication may be reproduced in any form whatsoever or used to make any derivative work without prior written approval by Coordinate Co., Ltd.

No representation of warranties for fitness for any purpose other than what is specifically mentioned in this guide is made by Coordinate Co., Ltd.

Coordinate Co., Ltd. reserves the right to revise this publication and make changes without obligation to notify any person of such revisions or changes. Coordinate Co., Ltd. may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time. If there is any software on removable media described in this publication, it is furnished under a license agreement included with the product as a separate document. If you are unable to locate a copy, please contact Coordinate Co., Ltd. and a copy will be provided to you. Unless otherwise indicated, Coordinate registered trademarks are registered in the People's Republic of China and other territories. All registered trademarks recognized.

For further information please contact Coordinate Co., Ltd. Local Office.

Product Version 1.1, Apr 2005

Publication 1

<http://www.coordinate.com.cn>

## Content

AT91RM9200 BSP Programmer Guide .....	1
Content .....	2
About this Manual .....	3
Related Documents .....	3
Overview .....	4
Introduction .....	4
Version .....	4
Design Consideration .....	4
Product and its Enviroment .....	5
Glossary of Terms .....	6
Using the Product .....	7
Product Concepts .....	7
Product Install .....	7
Product Compile .....	7
Memory Mode Debug .....	7
Demo Usage .....	8
Modification Notes .....	9
FLASH .....	9
SDRAM .....	9
CLOCK .....	9
TC .....	10
AIC .....	10
UART .....	10
EMAC .....	11
Contacting Coordinate Support .....	12

## About this Manual

这份 Programmer Guide 主要是用于帮助客户全面细致深入地了解 AT91RM9200 BSP，使得客户得以更加快速的理解和开始使用 AT91RM9200 BSP，从而缩短 Time to Market。

Programmer Guide 中包括了如下几个部分：

- **Overview**  
介绍 AT91RM9200 BSP，及其设计思想，还有该 BSP 如何使用于应用环境中，包括与操作系统 vxWorks 和上层应用程序之间的关系
- **Using the Product**  
详细介绍产品的安装步骤和编译过程，并通过示范用例帮助客户更快地了解如何使用 AT91RM9200 BSP。  
并且介绍如何使用本产品，编译出用于调试新硬件的内存模式 Bootrom
- **Contacting Coordinate Support**  
科浩鼎信息技术有限公司技术支持的联系方式。

## Related Documents

- **AT91RM9200 BSP Release Notes (PDF Format)**
- **vxWorks BSP Developer's Guide (PDF Format)**
- **vxWorks Drivers API Reference (PDF Format)**
- **vxWorks Programmer Guide (PDF Format)**

## Overview

### Introduction

AT91RM9200 是 ATMEL 公司针对工业应用领域推出的一款基于 ARM920T 具有工业级芯片的控制器，提供了丰富的外围接口和可扩展 IO。

AT91RM9200 BSP 是一套针对 AT91RM9200 开发参考板研制的，应用于实时操作系统 vxWorks 5.5 上的 BSP。

AT91RM9200 BSP 提供了众多的宏定义和数据结构，方便用户灵活的根据系统和主控板硬件的变化重新配置 BSP。

注：以下 AT91 即为 AT91RM9200 的简称

### Version

AT91RM9200 BSP 当前版本为 v1.1，适用于美国风河公司的实时操作系统 vxWorks 5.5。

### Design Consideration

如下的这些设计考虑是为了充分发挥 AT91RM9200 上各个控制器的性能，并方便用户简单快捷的使用 BSP 生成自定义的 Bootrom 和 vxWorks 模式：

1. 完成各 I/O 设备和 END 设备的挂接  
为了方便客户开发应用程序，同时屏蔽相关的硬件信息，AT91RM9200 BSP 在初始化过程中能够自动 I/O 设备和 END 设备的挂接，使得上层应用程序在 vxWorks 启动完成以后可以直接使用标准的 I/O 接口和网络协议栈提供的报文和数据收发等接口进行数据通信和控制。
2. 用于调试新硬件的内存模式 Bootrom  
考虑到用户用于存放 Boot Code 的硬件可能是贴片的 Flash，而不是 Socket 型的 eeprom，如果在 BSP 初期调试阶段需要反复修改程序的情况下重复烧写 Flash 比较浪费开发时间，因此提供了在内存中仿真 Flash 地址空间存放 Bootrom 并进行调试的内存调试模式。在该模式中 Bootrom 的起始地址将位于实际 RAM 的后半段地址空间，LOCAL\_MEM\_SIZE 将相应缩小。
3. 由于 UART 控制器在进行一定配置后，可作为红外、RS485、直接连接 Modem 等工作模式，而 DBGU 则没有这些功能，只能工作在 UART 模式下，因此需要将 UART 驱动和 DBGU 驱动分离开。

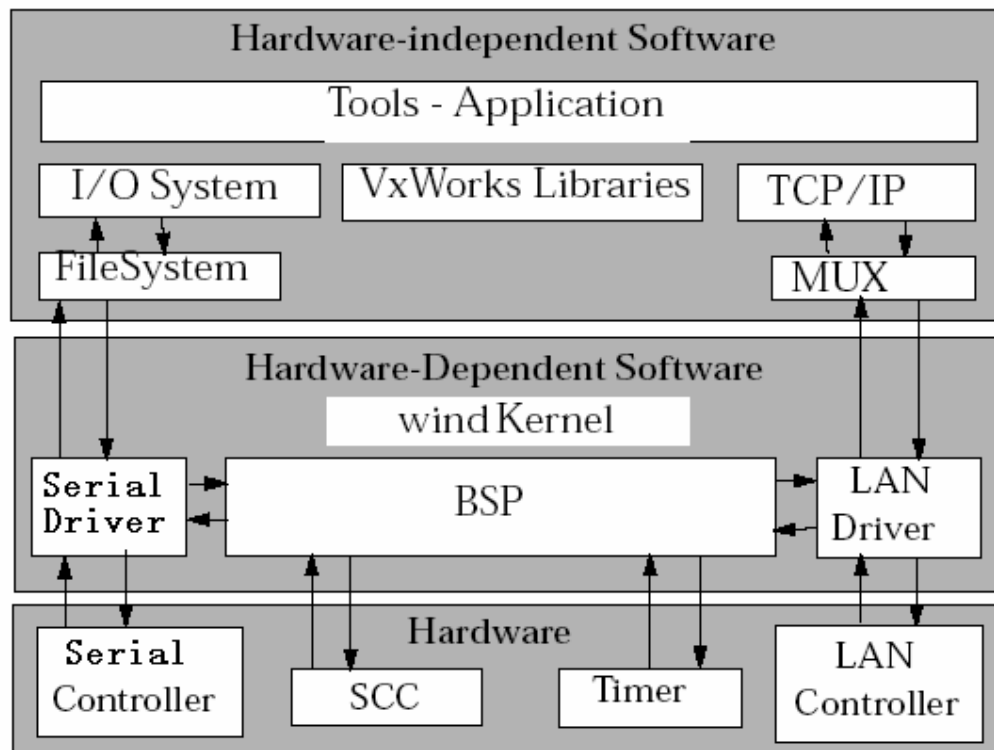
## Product and its Enviroment

从系统运行框架来看，AT91RM9200 BSP 运行在底层硬件设备和 vxWorks 操作系统之间，然后通过标准的 I/O 接口和网络协议栈提供的报文和数据收发等接口向上层应用程序提供数据通信和控制通道。

在 vxWorks 初始化过程中能够自动完成将串口设备向 I/O 设备层挂接，以及以太网设备向 END 设备层挂接

下图是 vxWorks 操作系统中的系统框架图。

## BSPs and VxWorks



## Glossary of Terms

**vxWorks:** 美国风河公司提供的实时操作系统

**Tornado:** 美国风河公司提供的面向 vxWorks 应用程序开发的集成开发环境

**AT91RM9200:** ATMEL 公司面向工业控制领域的主控芯片，内核是 ARM920T

**END:** 增强型网络设备

## Using the Product

### Product Concepts

AT91RM9200 BSP 即可以通过建立 Project，然后在 Project 环境下编译运行，也可以采用 Tornado 集成开发环境中菜单中的 Build 选项进行编译运行

以下将示范如何将 AT91RM9200 BSP 放置在 Project 环境下编译运行

### Product Install

安装 Tornado2.2 / vxWorks 5.5，或者找一台已经安装好的主机

1. 关闭 Tornado
2. copy {BSP} 目录中所有文件到 {WIND\_BASE} / target / config / at91dev 目录下
3. 启动 Tornado
4. 在 WorkSpace 环境下，选择 New Project
5. 选择 Create a bootable VxWorks image
6. 在 Step1 中填入相关项目名称、目录等
7. 在 Step2 中左侧选中“A BSP”，并在右侧下拉框中选择 at91dev
8. 选中下一步即可
9. 在 Properties of Project 中，选择 EXTRA\_MODULES，在 Value 中根据编译器类型添加相应的 at91DBGU.o, at91End.o, at91IntrCtrl.o, at91Timer.o, at91UART.o

### Product Compile

1. 在编译之前，请根据需要使用到的编译器类型（diab 或 gnu）从 drv.obj 目录下的相应目录中拷贝 at91DBGU.o, at91End.o, at91IntrCtrl.o, at91Timer.o, at91UART.o 至 drv.obj 目录下
2. 标准环境下，在 Tornado 菜单 Build 下选择 Standard BSP Builds，并再分别选择 BSP 为 at91dev 和 Bootrom.bin 或 vxWorks，最后单击 OK Button 即可。
3. Project 环境下，在 vxWorks Item 上右击，选择 Rebuild All(vxWorks)。

### Memory Mode Debug

在 ATMEL 公司的 AT91RM9200 芯片中，内置有 128KB 的 PROM，其中固化了一段初始化程序，当系统配置为从内部启动时，AT91 将顺序搜索 SPI 接口上的 DataFlash，TWI 接口上

的 EEPROM 和外围接口上的并行 8Bit Flash，如果这些 Flash 上的头部 32 个字节满足一定的内容要求，则从该设备启动，否则将通过 DBGU 或者 USB 下载启动程序至 SRAM 中，选择下载程序到 SRAM 中的原因是 SRAM 控制器的配置是固定的，而 SDRAM 控制器的配置是根据客户硬件设计变化的。具体说明参见 AT91 DataSheet 中 Boot Program 一节

当我们在进行 Bootrom 调试时，如果需要基于 Flash 进行调试，需要首先将 Bootrom 烧写至 Flash 中。这一过程如下：从内部 PROM 启动，通过串口的 xmodem 协议下载 Loader.bin 至 SRAM 中运行，Loader.bin 通过串口的 xmodem 协议下载 uBoot.bin 至 SDRAM 中运行，最后由 uBoot.bin 通过串口的 kermit 协议下载需要烧写的 Bootrom 至内存中并烧写至 Flash 中。这一过程步骤复杂，而且由于通过串口下载程序，速度较慢，因此不适合在调试 BSP 这一需要反复更改程序并烧写 Flash 的过程中使用。

由此，为了方便大家进行 Bootrom 调试，特别提供了内存调试模式，在 BSP 中通过宏定义开关 AT91\_RAM\_DEBUG 来选择是否使用 Bootrom 内存调试模式。在该模式下，ROM\_BASE\_ADRS 被定义为 0x20800000，即内存起始地址偏移 8MB 的位置（如果需要可以调整为 0x20400000 等等），LOCAL\_MEM\_SIZE 相应的调整为 0x800000（或者是 0x400000），这些宏需要在 Config.h 和 Makefile 中同步修改（这两个文件中的宏定义必须保持一致）；然后在 Tornado 环境下选择编译 bootrom/bootrom\_uncmp，再通过仿真器下载 bootrom.bin/Bootrom\_uncmp.bin 到 0x20800000（或者 0x20400000）处，由于在编译过程中已经通过 RAM\_HIGH\_ADRS 将 Symbol 指向 0x20200000，所以在运行至 usrInit 函数入口时就可以直接看到调试信息。在 bootrom/bootrom\_uncmp 运行至 bootInit.c 中的 romStart 函数后，bootrom 将被解压缩/Bootrom\_uncmp 将被拷贝至 RAM\_HIGH\_ADRS（symbol usrInit 指向该位置），然后跳转至该地址开始运行 usrInit。从这里开始就与从 Flash 中运行 Bootrom 并跳转至该位置完全一模一样了。

需要注意的是，在该模式中由于需要使用 SDRAM 存放 Bootrom，因此我们必须在 Load Bootrom 到 SDRAM 之前，通过仿真器的脚本对 AT91 进行配置，配置的内容包括：AT91 主时钟、AT91 外部时钟、EBI 控制器、SMC 控制器、SDRAMC 控制器。同时在 BSP 中通过宏定义开关 AT91\_RAM\_DEBUG 在 romInit.s 中禁止对这些时钟和控制器进行配置。

## Demo Usage

由于本 BSP 完全基于 vxWorks BSP Template Kit 开发，几乎所有函数和宏的说明与帮助可以在 vxWorks 相关帮助中找到，具体内容请参考 vxWorks BSP Developer's Guide 和 vxWorks Programmer Guide。



## Modification Notes

### FLASH

对于 Flash，需要根据 Flash 所使用的片选在 EBI 控制器中将对应的 EBI\_CSA 寄存器中的 CSnA 位进行配置，然后根据 Flash 的位宽、访问时序等信息配置相应的 SMC\_CSRn 寄存器，所有这些配置在 romInit.s 中进行。需要注意如果在 config.h 中定义了 AT91\_RAM\_DEBUG，则 romInit.s 中的这段配置程序将不被运行。其他需要配置的还有 config.h 中的 ROM\_BASE\_ADRS，ROM\_SIZE 等，需要注意这些宏的定义必须与 Makefile 中的一致。

### SDRAM

对于 SDRAM，要求将 EBI 控制器中将对应的 EBI\_CSA 寄存器中的 CS1A 位配置为 1，然后根据 SDRAM 的访问时序、行列地址线和刷新频率等信息配置相应的 SDRAMC 中的寄存器，所有这些配置在 romInit.s 中进行。需要注意如果在 config.h 中定义了 AT91\_RAM\_DEBUG，则 romInit.s 中的这段配置程序将不被运行。其他需要配置的还有 config.h 中的 RAM\_LOW\_ADRS，RAM\_HIGH\_ADRS，LOCAL\_MEM\_LOCAL\_ADRS，LOCAL\_RAM\_SIZE 等，需要注意这些宏的定义必须与 Makefile 中的一致。

### CLOCK

对于时钟，本产品所基于开发板的 AT91 主晶振为 18.432MHz，为了达到 180MHz 的 CPU 主时钟频率，需要在 CKGR\_PLLAR 寄存器中将主晶振时钟输入先除 4 然后倍 39 倍频，则可得 179712000Hz 的主时钟频率。然后为了向各个外设控制器，包括 SDRAMC 和 UART 等提供时钟，需要在 PMC\_MCKR 寄存器中通过将主时钟频率除 3 得到 59904000Hz 的外设时钟频率。具体计算解释可以参看 at91.h 中的说明。

这些系统时钟的配置也在 romInit.s 中完成，而且必须在初始化 SDRAM 控制器之前完成。需要注意如果在 config.h 中定义了 AT91\_RAM\_DEBUG，则 romInit.s 中的这段配置程序将不被运行。

## TC

对于 TC, vxWorks 要求提供两个独立的时钟用于系统时钟和辅助时钟。系统时钟用于实现 vxWorks 系统中的时间片 (用于任务延迟, 同优先级任务间轮转等)。辅助时钟用于提供高速高优先级的周期性中断 (可用于高速查询和统计等)。

在 AT91 中有两套 TC 控制器, 每个 TC 控制器控制 3 个 TC 通道。每个 TC 通道都有独立的时钟输入, 计数器和信号输出。在本实现中使用了第零套 TC 控制器中的 TC0 (系统时钟) 和 TC1 (辅助时钟), 时钟输入均采用外设时钟频率除 32 得到 1872000Hz。如果需要使用 TC, 请使用 TC2, 或第一套 TC 控制器中的 TC0/1/2。

## AIC

对于 AIC, 本系统提供了对于各个外设控制器和外部中断输入的 INT\_LVL 的定义 (在 at91rm9200.h 中定义, 用于 intConnect 和 intEnable)。

同时由于 AT91 中的 AIC 提供了中断优先级机制, 即高优先级中断可以打断低优先级中断, 高优先级中断可以屏蔽低优先级中断, 因此提供了中断优先级的设定, 在 sysLib.c 中有一个数组 at91IntLvlPriMap, 其中每一项依次定义了各个中断的优先级, 具体顺序和 ID 号请参考 AT91 的 DataSheet 中的 Peripheral Identifiers 定义。

## UART

在 AT91 中有 DBGU 和 UART 两种 UART 控制器, 前者只能工作在 UART 模式, 而后者还可以工作于 IrDA、ISO7816、RS485 和 Modem 模式。因此分别实现这两个控制器的驱动程序。

本实现将自动把 DBGU 和 UART 控制器作为 tty 设备挂接至 tty 层, 再通过 ttyDrv 挂接到 IO 层。用户只需要修改相应的控制器硬件配置参数。具体的配置在 sysSerial.c 中的 at91UartParas 结构数组中定义, 其中的每一项依次对应于 DBGU、UART0、UART1、UART2、UART3。由于作为 Console 的 DBGU 端口用于调试开发是必须的, 因此在 at91.h 中定义 N\_SIO\_CHANNELS 为 (N\_UART\_CHANNELS + 1), 其中的 1 就是指 DBGU, 而 N\_UART\_CHANNELS 则根据具体实现需要使用的 UART 数目而定, 在定义了 N\_UART\_CHANNELS 后, 需要在 sysSerial.c 中通过宏定义开关选择使能相应 UART 控制器的配置参数结构。

UART 的配置参数结构需要根据实际情况修改前 6 个参数, 分别是:

- PIOA 的外设 A 选择
- PIOA 的外设 B 选择
- PIOB 的外设 A 选择
- PIOB 的外设 B 选择
- UART 波特率
- 硬件配置选项 (包括数据位数、停止位数、奇偶校验等)

## EMAC

对于 EMAC，在驱动程序已经完成了以太网控制器的初始化，根据 PHY 自动协商结果进行工作模式配置、以太网报文的发送和接收、中断的使能和停止、组播地址的添加和删除等功能。因此在修改时只需要根据实际硬件实现修改：

- sysLib.c 中 sysAt91EmacEnetEnable 函数的实现（根据具体使用的硬件引脚定义）
- sysAt91EmacEnd.c 中发送和接收 BD 的数目（AT91\_EMAC\_TBD\_NUM 和 AT91\_EMAC\_RBD\_NUM）和 BD 的基址寄存器（AT91\_EMAC\_BD\_BASE）
- at91End.h 中的 PHY\_MODE\_MII 定义（MII 还是 RMII）

这些配置信息将在 sysAt91EmacEndLoad 中被组织成 paramStr 传入 EMAC 控制器的初始化函数 at91EmacEndLoad 中。

## Contacting Coordinate Support

Coordinate Co., Ltd.  
Beijing Office  
Tel: (+86)-10-64205651  
Fax: (+86)-10-64205652

Coordinate Co., Ltd.  
Shanghai Office  
Tel: (+86)-21-52681026  
Fax: (+86)-21-52681027

[Eric.Zha@coordinate.com.cn](mailto:Eric.Zha@coordinate.com.cn)

When you contact Coordinate technical support, please write the following in the Subject field of your e-mail: "question" and a summary/headline describing the problem, thanks.

Coordinate Co., Ltd. website <http://www.coordinate.com.cn>