



**H-JTAG**

**北京中科凌创电子**

# **使用 H-JTAG 一键烧写嵌入式系统**

文档版本 A

发布日期 2013-01-10

**[WWW.HJTAG.COM](http://WWW.HJTAG.COM)**

---

## 使用 H-JTAG 一键烧写嵌入式系统

版权所有 © 2013 H-JTAG 北京中科凌创电子科技有限公司

### 修改记录

日期	版本	改动
2013-01-10	A	发布第一版本

### 版权声明

1. 文档中提及的任何第三方的注册商标和产品标识，均属于第三方公司所有；
2. 如果文档当中有任何地方侵犯了您的权利和版权，请和我们联系，我们将及时修改；
3. 本文档为开放文档，用户可以在保证文档完整性的前提下，自由分发。

### 官方主页

[www.hjtag.com](http://www.hjtag.com)

### 技术支持

[support@hjtag.com](mailto:support@hjtag.com) / [twentyone@hjtag.com](mailto:twentyone@hjtag.com)

### 销售咨询

[sales@hjtag.com](mailto:sales@hjtag.com)

## 使用 H-JTAG 一键烧写嵌入式系统

H-JTAG 自带的烧写软件 H-FLASHER 提供了强大灵活的 NAND FLASH 烧写支持. 本文详细介绍了如何使用 H-JTAG/H-FLASHER 实现 NAND FLASH 嵌入式系统一键烧写, 并结合实例说明了具体的实现步骤.

由于 NAND FLASH 低价格, 高容量的特点, 在嵌入式系统设计中得到越来越多的使用. 很多流行的硬件平台, S3C2410/S3C2440/S3C6410/PXA300 等, 都提供了 NAND FLASH 支持. 在嵌入式应用中, 应用最广泛的操作系统包括 LINUX 和 WINCE. 这两种常见的操作系统都包括了启动程序, 内核镜像和文件系统等多个文件. 在产品量产烧写时, 如果能实现 NAND FLASH 一键烧写, 一个步骤完成多个文件的烧写, 生产效率能够得到有效的提高.

下面, 将详细介绍如何使用 H-JTAG/H-FLASHER 实现 NAND FLASH 一键烧写. 为了方便说明, 后面的介绍将以 S3C2440+K9F128 测试平台为实例, 一步一步介绍如何实现 LINUX 的一键烧写. 虽然文中以 S3C2440+K9F1208 为实例加以介绍, 但 NAND FLASH 一键烧写的实现, 可以应用到不同的硬件和嵌入式操作系统平台下.

### 1. 准备工作

首先, 用户需要先准备一块已经烧写好的嵌入式系统的目标系统. 在我们使用的 S3C2440+K9F1208 测试平台中, 已经事先成功烧写了 LINUX 系统. 然后, 将 H-JTAG USB 仿真器和目标系统链接好. 连接好后, 给目标系统上电. 接下来运行 H-JTAG 软件. H-JTAG 启动后, 会自动执行 JTAG 检测, 并在成功检测到 CPU 后在主界面上显示其 JTAG ID 和 ARM 内核. 以 S3C2440 为例, H-JTAG 检测到 S3C2440 后, 主界面显示如下图所示.

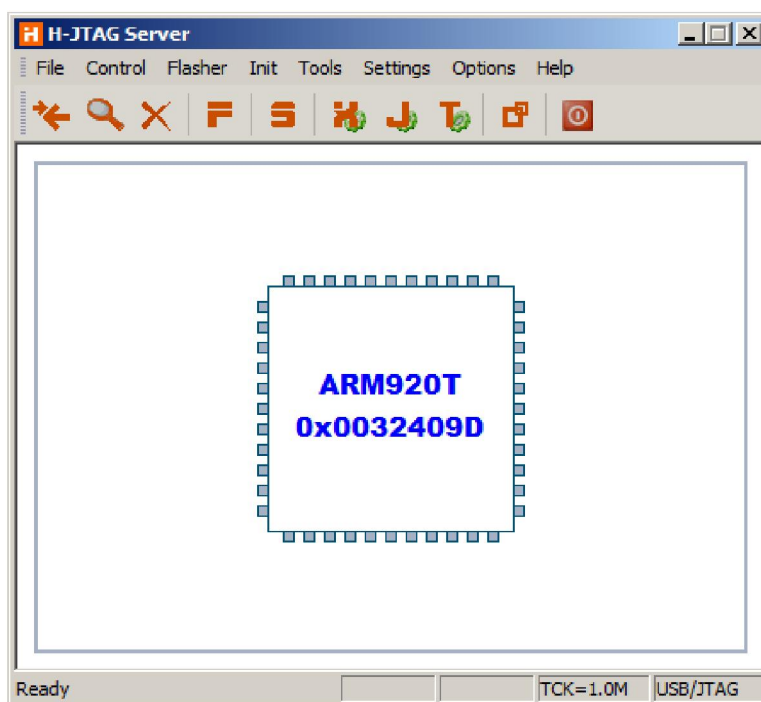


图 1: 目标系统检测

## 2. 配置 H-FLASHER

在运行 H-JTAG 软件后, 接下来需要运行 H-FLASHER 烧写软件, 并针对目标系统作些基本配置. 对于常见的硬件平台, H-JTAG 提供了 H-FLASHER 参考配置文件, 其存放路径为: "H-JTAG 安装路径\HFC Examples". 用户可以直接在 H-FLASHER 中装载相应的 HFC 配置文件. 如果没有参考配置, 用户可以自行配置. 关于 H-FLASHER 的具体使用, 请参考<<H-JTAG USER MANUAL>>.

因为 H-JTAG 提供了针对 S3C2440+K9F1208 的配置文件, 我们在测试中可以直接在 H-FLASHER 中装载 S3C2440+K9F1208.HFC 配置文件. 装载后, H-FLASHER 主界面如下图所示:

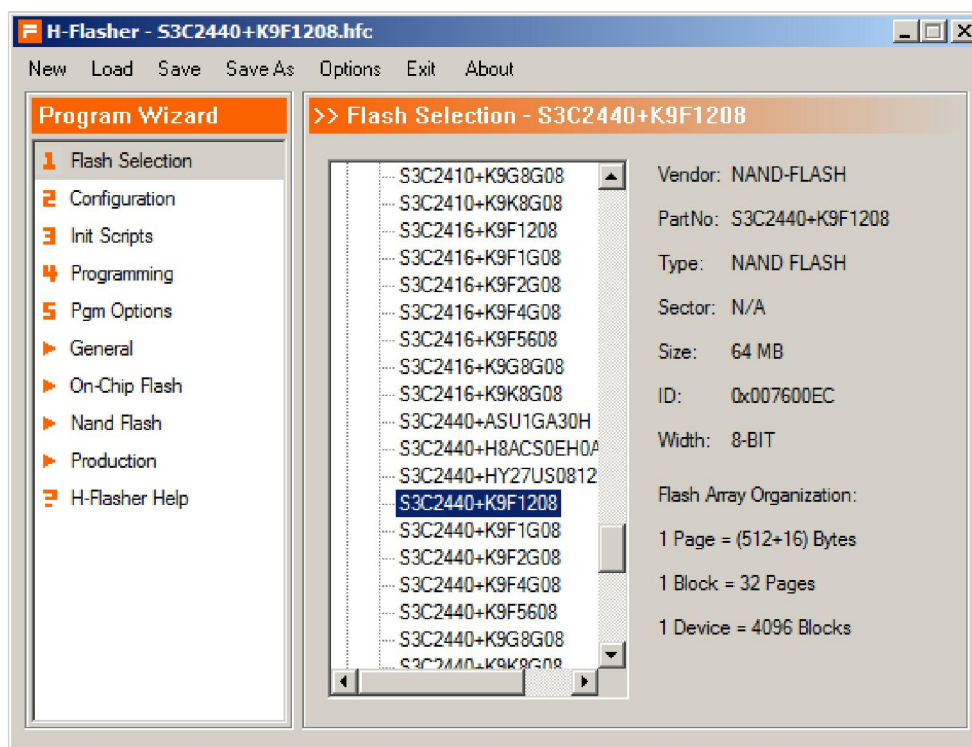


图 2: 装载 HFC 配置文件

## 3. 复制 NAND FLASH

接下来, 用户需要将目标系统中已经烧写的嵌入式系统复制出来, 以获取一个包含完整系统的独立文件. H-FLASHER 提供了 Dump 操作. 通过 Dump 操作, 用户可以将 NAND FLASH 中指定范围的数据给复制出来. Dump 操作会读取的数据包括指定范围内每个页面的主数据区和空闲数据区的数据.

**在 Dump 的时候, 不要选择整片 FLASH. 一是, FLASH 最后面的一些 BLOCK, 一般都未使用, 不需要 Dump 整片 FLASH. 二是, 如果选择 Dump 整片 FLASH, 要将 Dump 回来的文件烧写到含有坏块的 FLASH 中去的时候, 会提示文件太大, 导致无法烧写.**

在我们使用的测试平台中, U-BOOT, LINUX 内核镜像和 LINUX 文件系统占用了 K9F1208 从 Block-0/Page-0 到 Block-3850/Page-31 的范围空间. 所以, 我们只需要将这段空间的数据 Dump 回来即可. 具体设置如下图所示:

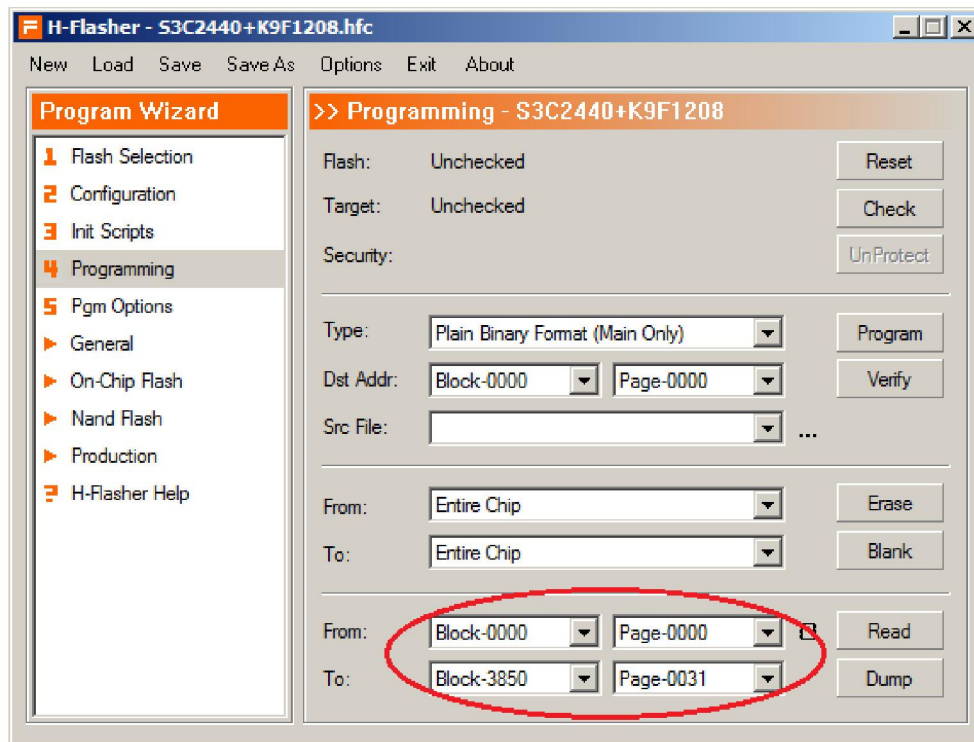


图 3: Dump 设置

设置好后, 点击 **Dump** 按钮, H-FLASHER 会先提示让用户选择将文件保存到那里, 然后开始从 FLASH 中 Dump 数据. Dump 过程中, H-FLASHER 会实时显示时间, 速度和进度. 完成后, 提示 Dump 完成, 如下图所示.

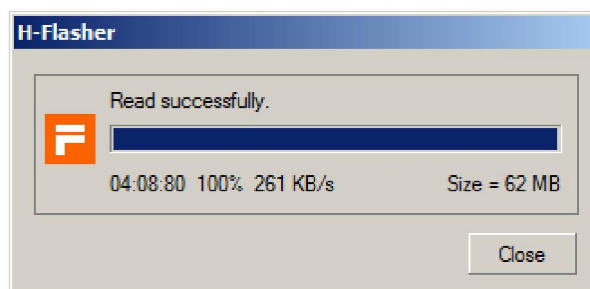


图 4: Dump 文件

在 Dump 完成后, 用户就得到了一个包含完整系统的独立文件. 以我们使用的测试平台为例, 得到的文件包含了 U-BOOT, LINUX 内核镜像和 LINUX 文件系统. 此外, 该文件还包含了每个页面中空闲数据区的数据, 主要是 OOB/ECC 数据.

## 4. 一键烧写

通过系统复制, 用户已经成功获得一个包含完整系统的独立文件. 在产品量产烧写的时候, 通过烧写该文件即可实现一键烧写.

假设用户获取的独立文件的名称为 Dump.bin. 配置 H-FLASHER 时, 类型选择为 Plain Binary Format (Main + Spare/Oob), 地址设置为 Block-0/ Page-0, 烧写文件选择 Dump.bin. 设置如下图所示.

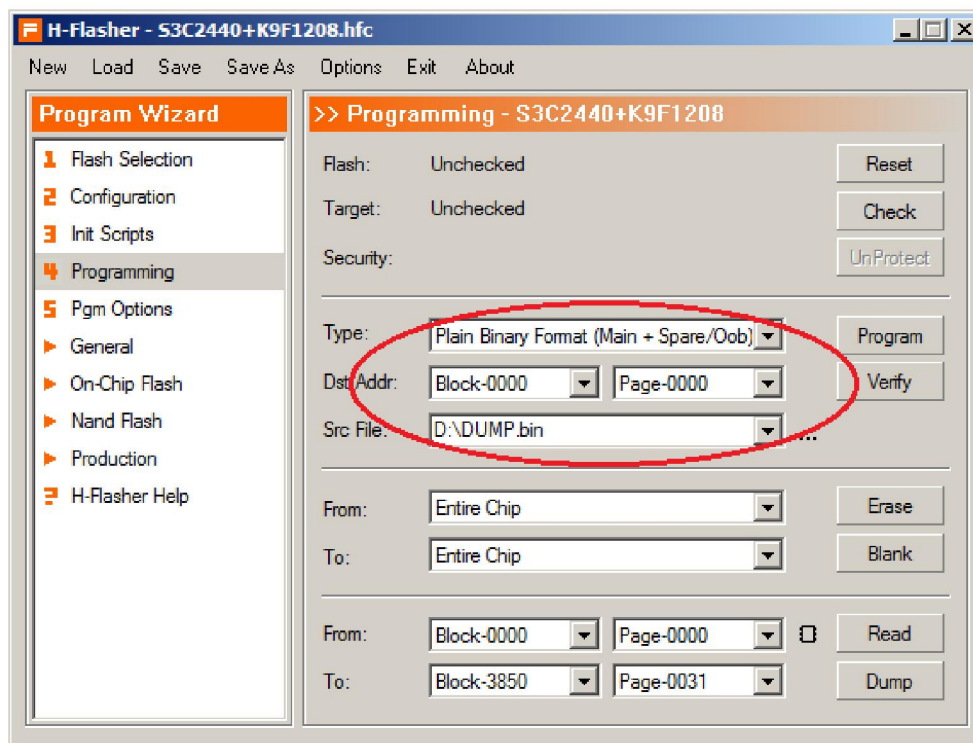


图 5: 一键烧写设置

设置好后, 点击 Program 按钮, H-FLASHER 就会自动开始烧写. 烧写过程中, H-FLASHER 会实时显示时间, 速度和进度. 烧写完成后, 提示烧写成功, 如下图所示. 然后, 给目标系统重新上电, 系统即可正确运行.

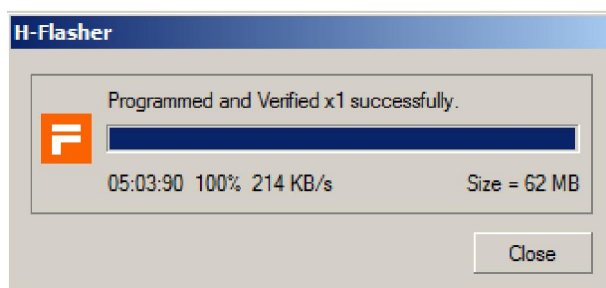


图 6: 一键烧写

## 5. 产品烧写模式

此外，H-FLASHER 还支持产品烧写模式。在 H-FLASHER 左侧的编程导航中点击 Production，就会进入烧写模式设置页面。在该页面中，选择“Enable production mode for flash programming”即可使能产品烧写模式。

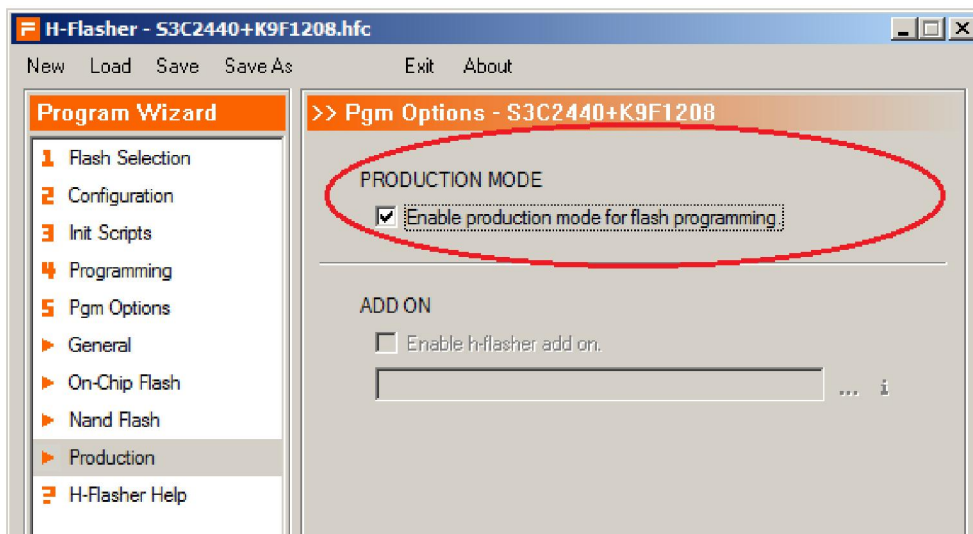


图 7: 使能产品烧写模式

使能产品烧写模式后，在编程页面上点击 Program 按钮，H-FLASHER 就会自动进入产品烧写模式，如下图所示。

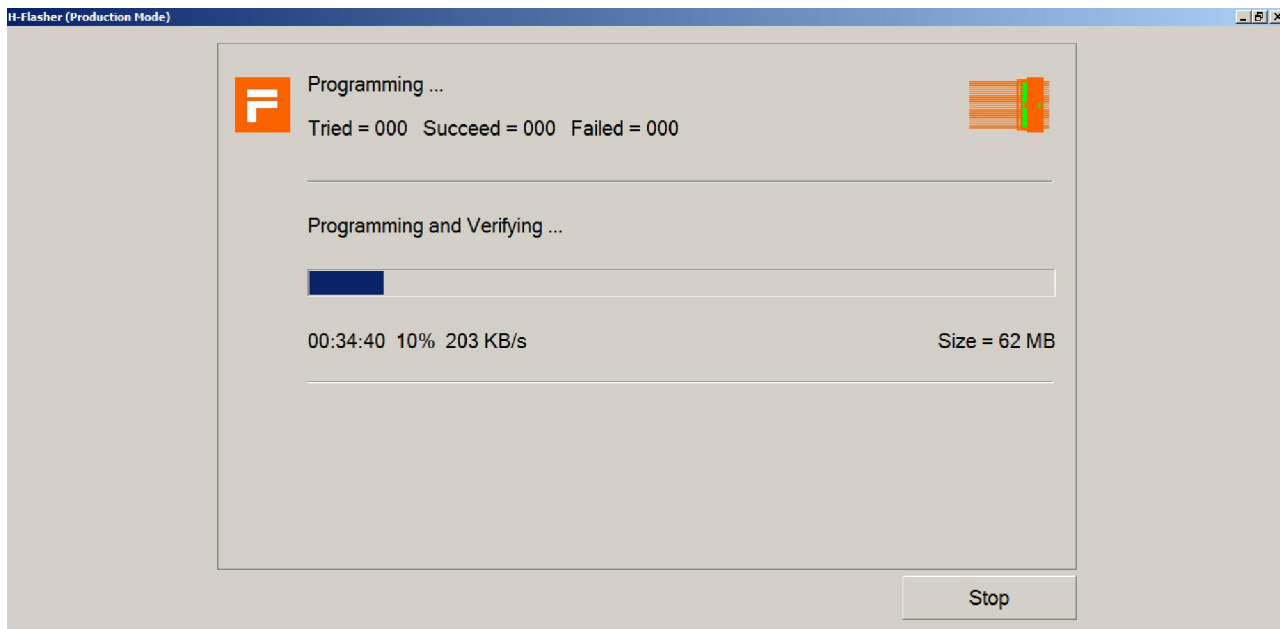


图 8: 产品烧写模式

在产品烧写模式下，用户操作被简化，H-FLASHER 完全通过检测 JTAG 的插拔来自动控制烧写，以提高效率。当 H-FLASHER 检测到有新的连接时，自动开始烧写。烧写完成后，提示用户断开当前连接。H-FLASHER 检测到连接断开后，会进入新的循环，等待下一个新连接。整个过程都由 H-FLASHER 自动控制，用户无需在 PC 端做任何鼠标/键盘操作。