

Uniflash 和 Daisy Chain 在芯片量产中的应用

Holly Gu

Sales and Marketing/East China

ABSTRACT

随着设计水平的提高，工业客户的系统越来越复杂，在系统的单板设计中常常需要多个嵌入式处理器协同工作。为了节省成本，提高量产效率，避免产线工人误操作，往往采用菊花链（Daisy Chain）结构来实现利用一个 JTAG (Joint Test Action Group) 口调试、更新多个处理器固件的目的。本应用报告详述了将同类型处理器的调试口以 Daisy Chain 形式连接并利用 Universal Flash Programmer (Uniflash) 工具的批处理方式下载固件的方法。在内容安排上，首先介绍了菊花链(Daisy Chain) 电路的基本原理，其次介绍了实验软硬件平台，最后介绍了基于 Daisy Chain 和 Uniflash 的量产批处理文件生成方法和执行结果。

Contents

1	Daisy Chain 简介	2
2	实验软硬件平台	2
	2.1 实验软件平台	2
	2.2 实验硬件平台	3
	2.2.1 RM57L843 及 XDS200 简介.....	3
	2.2.2 基于 RM57L843 的 Daisy Chain 硬件平台.....	3
3	基于 Daisy Chain 的量产批处理文件.....	4
	3.1 CCS Target 的配置.....	4
	3.2 .ccxml 配置文件解析.....	5
	3.3 利用 Uniflash 下载固件	6
	3.3.1 Load Image 方式.....	6
	3.3.2 批处理方式	7
	3.4 实验结果	9
4	参考文献	9

Figures

Figure 1.	Daisy Chain 硬件连接示意图	2
Figure 2.	基于 RM57L843 的 Daisy Chain 硬件连接示意图.....	3
Figure 3.	Daisy Chain 结构 RM57L843_Core0 配置示例	4
Figure 4.	Daisy Chain 结构 RM57L843_Core1 配置示例	5
Figure 5.	RM57L843_Core0.ccxml 配置文件.....	5
Figure 6.	RM57L843_Core1.ccxml 配置文件.....	6
Figure 7.	Load Image 方式下载固件	7
Figure 8.	批处理文件的产生方法.....	7
Figure 9.	dslite.bat 批处理文件脚本.....	8
Figure 10.	Uniflash_Core1 dslite.bat 脚本运行结果.....	9

1 Daisy Chain 简介

对于复杂的嵌入式系统，单个器件往往不能满足设计的需求，可能需要多个器件的协调配合才能完成用户设定的任务。对于多个器件，特别是 ARM 器件的调试，为了简化设计，可以将硬件连接成 Daisy Chain 的形式。

Daisy Chain 的应用需要 JTAG 设计规范的支持作为保障，这样在电路板上，仿真器只需要连接到其中一个 JTAG 端口，就可以对 Daisy Chain 上任意符合连接规范的芯片进行仿真调试。连接 Daisy Chain 所需要的引脚是：测试数据输入（TDI），测试数据输出（TDO），测试时钟（TCK），测试模式选择（TMS），测试复位（TRST）。其中，TDI 和 TDO 信号串联，TMS、TCK 和 TRST 信号并联。Daisy Chain 硬件连接示意图如 Figure 1 所示。

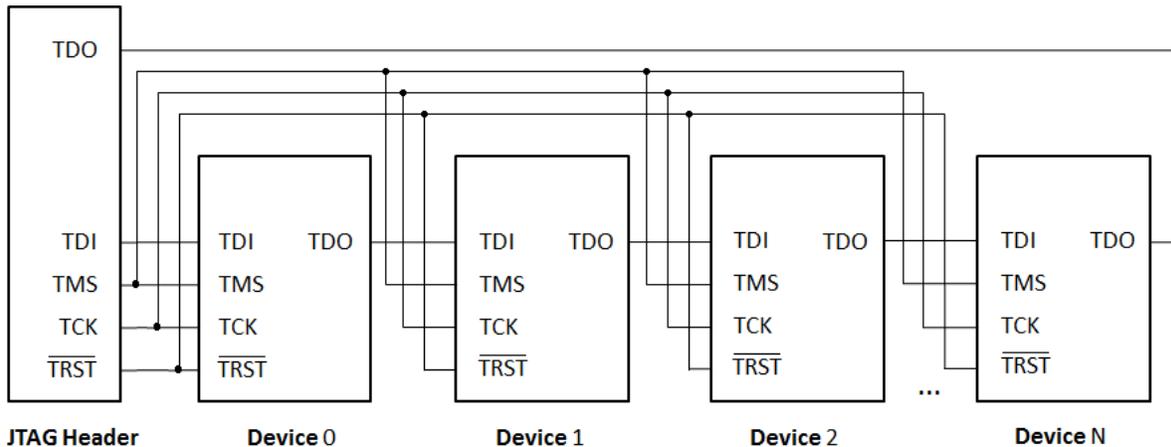


Figure 1. Daisy Chain 硬件连接示意图

2 实验软硬件平台

本节主要介绍实验的软硬件平台。软件平台由 Code Composer Studio (CCS) 和 Uniflash 构成。选取两片 RM57L843 芯片，即 RM57L843_Core0 和 RM57L843_Core1，来构建硬件实验平台。它们的 JTAG 以 Daisy Chain 的形式接入电路中。

2.1 实验软件平台

CCS 是德州仪器针对嵌入式处理器开发的集成仿真环境，基于 Eclipse 架构，包含有优化过的 C/C++ 编译器，编辑器，编译器，调试器等。Uniflash 是德州仪器提供的针对器件 Flash 进行操作的工具，可脱离 CCS 独立使用，界面友好，方便易用，特别适合在量产阶段将软件代码下载到目标板上。

本文中，在 CCS 中编辑一个非常简单的闪灯程序，配置 Daisy Chain 结构，生成相应的目标配置文件 .ccxml。Uniflash 利用 CCS 生成的 .ccxml 文件产生独立命令包（Stand along command line package）和可用于下载的批处理文件。CCS 的版本为 7.1.0，Uniflash 的版本 4.2.0，详细用法将会在第三节中介绍。

2.2 实验硬件平台

本文将 RM57L843_Core0 和 RM57L843_Core1 的 JTAG 以 Daisy Chain 的形式接入电路中，并采用 XDS200 仿真器来下载软件代码。

2.2.1 RM57L843 及 XDS200 简介

RM57L843 是德州仪器 ARM Cortex-R 内核 Hercules RM 系列的一员，它有两个 Cortex-R5F 内核形成互锁结构，主频可达 330MHz，1.66DMIPS/MHz。该器件还有具有单比特错误纠正和双比特错误检测功能的大容量 FLASH 和 RAM（4MB/FLASH, 512KB/RAM），以及诸如 10/100 Mbps 以太网 MAC（Ethernet MAC，EMAC），多缓存 ADC（Multibuffered Analog-to-Digital Converter，MibADC）等外设，丰富的设计资料，软件工具，以及可以提供对 IEC61508 功能安全的支持，非常适合工业应用。

XDS200 是德州仪器为处理器提供的新型 JTAG 仿真器，支持读/写内存，读寄存器，下载软件，实时仿真等功能。它是大家熟知的低成本仿真器 XDS100 和高性能仿真器 XDS560 的折中，具有较高的性价比。值得注意的是，如果用户使用的 CCS 是 5.2.x 及以前的版本，则需要在 CCS 里安装最新的插件来支持 XDS200，5.3.0 以后的 CCS 版本则不需要作特别处理。

2.2.2 基于 RM57L843 的 Daisy Chain 硬件平台

基于 RM57L843_Core0 和 RM57L843_Core1 的 Daisy Chain 硬件连接示意图如 Figure 2 所示。其中，TDI 和 TDO 信号串联，TMS，TCK，RTCK 和 nTRST 信号并联，RM57L843_Core0 的 TDI 作为 JTAG Header 的 TDI，RM57L843_Core1 作为 JTAG Header 的 TDO。同时所有的 JTAG 部分需要共享电源和地。为突出重点，本文仅给出了两片 RM57L843 的 JTAG 部分的示意图，用户可方便地应用到自己的设计中。

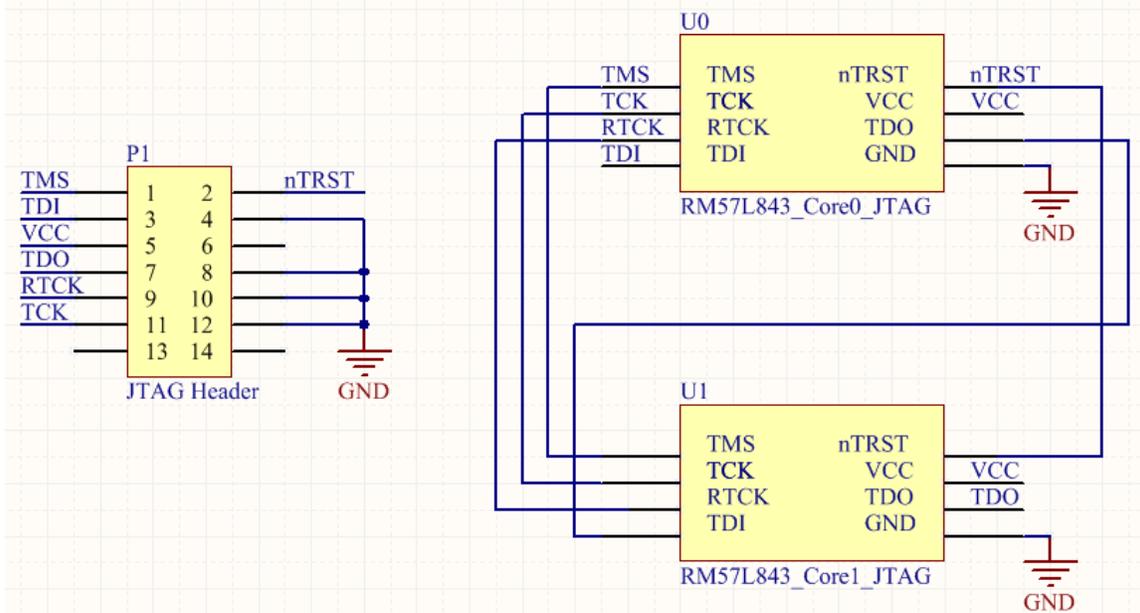


Figure 2. 基于 RM57L843 的 Daisy Chain 硬件连接示意图

本文以一个菊花链上存在两个 RM57L843 为例，对于多个芯片的拓扑可以类推扩展，此处不再赘述。

3 基于 Daisy Chain 的量产批处理文件

生成可用于量产的批处理文件的大致流程如下：用户通过 CCS 编辑编译出可实现功能的固件（.txt 格式），并配置 Target，生成配置文件.ccxml。Uniflash 加载.ccxml 文件产生独立命令包（Stand along command line package）和批处理文件。本章节通过一个具体实例来特别介绍如何生成适用于 Daisy Chain 量产批处理文件的方法。

3.1 CCS Target 的配置

在实际应用中，需要针对 RM57L843_Core0 和 RM57L843_Core1 分别创建工程，编写代码。在 CCS 中创建基于 RM57L843 工程的步骤此处不再赘述。在各自工程下的 TargetConfigs 目录中会有相应的 RM57L843_Core0.ccxml 和 RM57L843_Core1.ccxml 文件。双击.ccxml 文件，Advanced 选项是图形化的配置方式，Source 则是相应的脚本文件。

默认的配置是针对一个 JTAG 和单核的，如果要用作 Daisy Chain，需要做一些修改。在 All Connections 选项中列出了配置所需的关键项，用户可以通过右侧的菜单实现创建、增加、删除、调序、保存、导入、检测的功能。系统自动生成的配置文件中，一个 Debug Probe 只会自带一个器件（本例中 Core0），由于 Daisy Chain 上存在两个 JTAG 模块，配置 RM57L843_Core0 时，需要额外添加一个器件（本例中 Core1）。操作时先选中上一级模块，点击 Add 进入新增子模块配置界面，在跳出菜单中选择合适的器件。配置 RM57L843_Core1 模块时，只需调节 Up/Down，将 Core0 的位置调节到 Core1 后面。配置完成后，保存调试环境的配置文件，生成 Uniflash 所需的.ccxml 文件。Debug Probe 和器件的名字均可根据需要重新命名。

Figure3 是 RM57L843_Core0 采用 Daisy Chain 结构时的配置示例，Figure4 则是 RM57L843_Core1 在 Daisy Chain 结构下的配置示例。

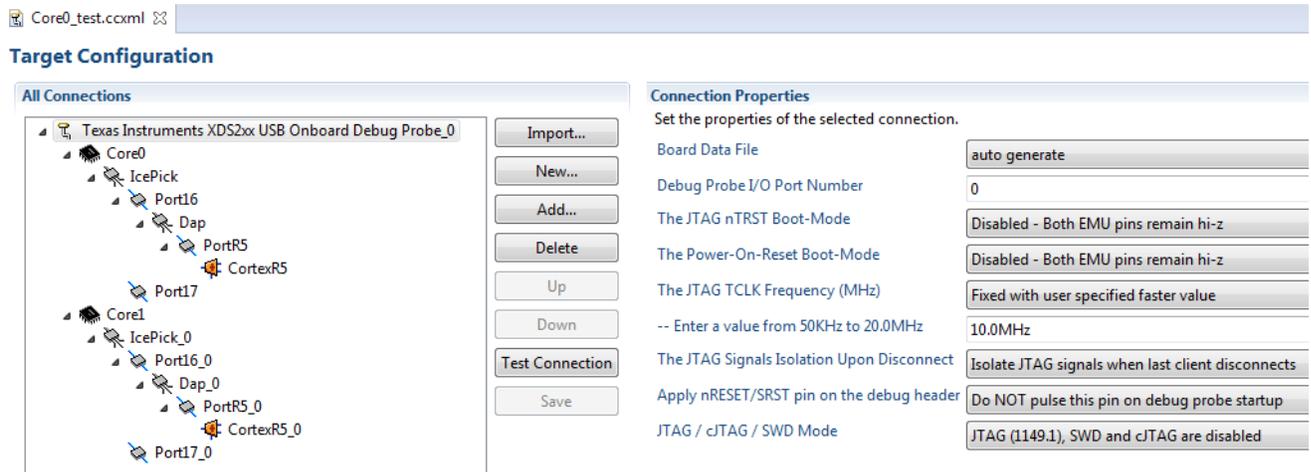


Figure 3. Daisy Chain 结构 RM57L843_Core0 配置示例

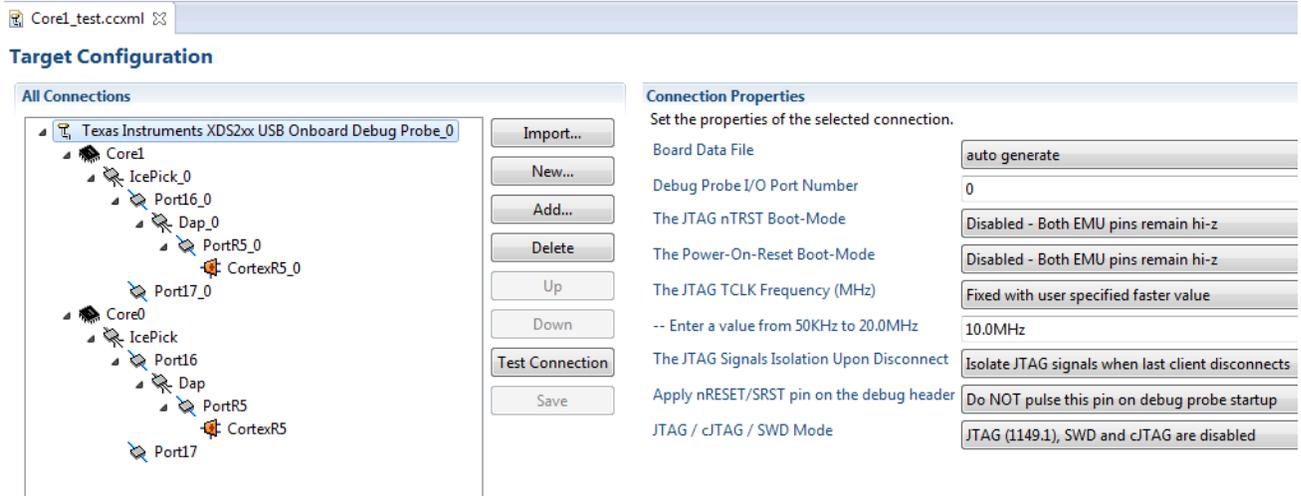


Figure 4. Daisy Chain 结构 RM57L843_Core1 配置示例

从 Figure 3 和 Figure 4 可以看出，JTAG 链上的两个 Core 在配置文件中的位置互为颠倒，这也是 Daisy Chain 配置的核心所在。

3.2 .ccxml 配置文件解析

Advanced 图形界面配置好 Target 后，CCS 会自动生成的 .ccxml 配置文件。该文件中包含仿真器，芯片内核，资源等信息，是 Uniflash 工具烧写芯片的依据。Figure 5 和 Figure 6 所示分别为 RM57L843_Core0 和 RM57L843_Core1 的 .ccxml 配置文件。方框中的部分是两个配置文件主要的不同。

```

1 on="1.0" encoding="UTF-8" standalone="no"?>
2 ions XML_version="1.2" id="configurations_0">
3 uration XML_version="1.2" id="configuration_0">
4 stance XML_version="1.2" desc="Texas Instruments XDS2xx USB Onboard Debug Probe" href="connections/TIXDS2XXUSB_onboard_Connection.xml" id="Texas
5 nnection XML_version="1.2" desc="Texas Instruments XDS2xx USB Onboard Debug Probe_0" id="Texas Instruments XDS2xx USB Onboard Debug Probe">
6 <instance XML_version="1.2" href="drivers/tixds560icepick_c.xml" id="drivers" xml="tixds560icepick_c.xml" xmlpath="drivers"/>
7 <instance XML_version="1.2" href="drivers/tixds560cs_dap.xml" id="drivers" xml="tixds560cs_dap.xml" xmlpath="drivers"/>
8 <instance XML_version="1.2" href="drivers/tixds560cortexR.xml" id="drivers" xml="tixds560cortexR.xml" xmlpath="drivers"/>
9 <platform XML_version="1.2" id="platform_0">
10 <instance XML_version="1.2" desc="RM57L8xx" href="devices/rm57l8xx.xml" id="RM57L8xx" xml="rm57l8xx.xml" xmlpath="devices"/>
11 <device HW_revision="1" XML_version="1.2" desc="Core0" description="RM57L8xx BGA, 4MB Flash, 512kB RAM, 10/100M Ethernet" id="RM57L8xx" partnum="
12 <instance XML_version="1.2" desc="Core1" href="devices/rm57l8xx.xml" id="Core1" xml="rm57l8xx.xml" xmlpath="devices"/>
13 <device HW_revision="1" XML_version="1.2" description="RM57L8xx BGA, 4MB Flash, 512kB RAM, 10/100M Ethernet" id="Core1" partnum="RM57L8xx"
14 <router HW_revision="1.0" XML_version="1.2" desc="IcePick_0" description="ICEPick_C Router" id="IcePick_C_0" isa="ICEPICK_C">
15 <subpath desc="Port16_0" id="Port16">
16 <router HW_revision="" XML_version="1.2" desc="Dap_0" description="CS_DAP Router" id="CS_DAP_0" isa="CS_DAP">
17 <subpath desc="PortR5_0" id="PortR5">
18 <cpu HW_revision="" XML_version="1.2" desc="CortexR5_0" description="Cortex_R5 CPU" deviceSim="false" id="Cortex_R5_0"
19 </subpath>
20 </router>
21 </subpath>
22 <subpath desc="Port17_0" id="Port17"/>
23 </router>
24 </device>
25 </platform>
26 onnection>
27 uration>
28 tions>
    
```

Figure 5. RM57L843_Core0.ccxml 配置文件

```

Core1_test.ccxml
1 n="1.0" encoding="UTF-8" standalone="no"?>
2 ons XML_version="1.2" id="configurations_0">
3 riation XML_version="1.2" id="configuration_0">
4 tance XML_version="1.2" desc="Texas Instruments XDS2xx USB Onboard Debug Probe" href="connections/TIXDS2XXUSB_onboard_Connection.xml" id="Texas
5 nnection XML_version="1.2" desc="Texas Instruments XDS2xx USB Onboard Debug Probe_0" id="Texas Instruments XDS2xx USB Onboard Debug Probe">
6 <instance XML_version="1.2" href="drivers/tixds560icepick_c.xml" id="drivers" xml="tixds560icepick_c.xml" xmlpath="drivers"/>
7 <instance XML_version="1.2" href="drivers/tixds560cs_dap.xml" id="drivers" xml="tixds560cs_dap.xml" xmlpath="drivers"/>
8 <instance XML_version="1.2" href="drivers/tixds560cortexR.xml" id="drivers" xml="tixds560cortexR.xml" xmlpath="drivers"/>
9 <platform XML_version="1.2" id="platform_0">
10 <instance XML_version="1.2" desc="Core1" href="devices/rm5718xx.xml" id="Core1" xml="rm5718xx.xml" xmlpath="devices"/>
11 <device HW_revision="1" XML_version="1.2" description="RM57L8xx BGA, 4MB Flash, 512kB RAM, 10/100M Ethernet" id="Core1" partnum="RM57L8xx" i
12 <router HW_revision="1.0" XML_version="1.2" desc="IcePick_0" description="ICEPick_C Router" id="IcePick_C_0" isa="ICEPICK_C">
13 <subpath desc="Port16_0" id="Port16">
14 <router HW_revision="" XML_version="1.2" desc="Dap_0" description="CS_DAP Router" id="CS_DAP_0" isa="CS_DAP">
15 <subpath desc="PortR5_0" id="PortR5">
16 <cpu HW_revision="" XML_version="1.2" desc="CortexR5_0" description="Cortex_R5 CPU" deviceSim="false" id="Cortex_R5_0" i
17 </subpath>
18 </router>
19 </subpath>
20 <subpath desc="Port17_0" id="Port17"/>
21 </router>
22 </device>
23 <instance XML_version="1.2" desc="RM57L8xx" href="devices/rm5718xx.xml" id="RM57L8xx" xml="rm5718xx.xml" xmlpath="devices"/>
24 <device HW_revision="1" XML_version="1.2" desc="Core0" description="RM57L8xx BGA, 4MB Flash, 512kB RAM, 10/100M Ethernet" id="RM57L8xx" partnum=
25 </platform>
26 nnection>
27 uration>
28 ions>

```

Figure 6. RM57L843_Core1.ccxml 配置文件

RM57L843_Core0 的配置文件中，Core0 为主，Core1 为从。RM57L843_Core1 的配置文件则相反。这样在 Daisy Chain 中，JTAG 工具可以通过识别配置文件来决定对哪个内核进行编程，也正是由于这些不同，Uniflash 工具才能够读取.ccxml 来实现烧写特定 CPU 的功能。

3.3 利用 Uniflash 下载固件

Uniflash 下载固件的方式有两种，加载 Image 方式和批处理方式。两种都可以支持 Daisy Chain 结构，本小节对两种方式分别作介绍。

3.3.1 Load Image 方式

使用 Load Image 时，首先需要创建一个 Session，点击 Session -> New Session，选择器件（Detected Devices），仿真器（Select Connection）和对应内核的.ccxml 文件。然后加载所需要下载的固件，如 Blink_core0.txt，再点击 Program 菜单中的 Load Image，即可将固件下载到目标板中。建立好 Session 后，下次使用时直接点击 Session -> Load Session 即可，不需要重复配置。Load Image 方式下载固件的方法如 Figure 7 所示。

针对 RM57L843_Core0 和 RM57L843_Core1，需要分别建立两个 Session，并执行两次 Load Image，用同一个仿真器在同一个 JTAG 口，将固件分别下载到 Core0 和 Core1 中。这种方法对 Daisy Chain 而言，由于有.ccxml 文件作引导，故不需要特别配置。

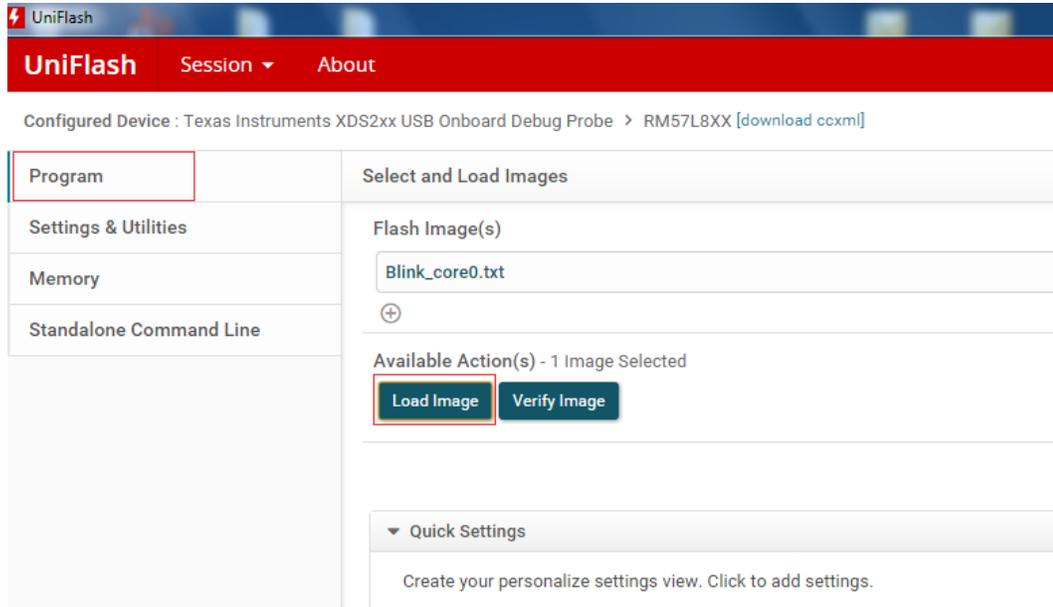


Figure 7. Load Image 方式下载固件

3.3.2 批处理方式

Load Image 的方式固然简单，但仍需要一定的步骤，对于产线上的工人来说，依然存在误操作的可能性。本小节介绍的批处理方式，只需双击鼠标就可实现固件的下载，简单方便。

3.3.2.1 批处理文件的产生

要实现批处理方式下载固件，首先要产生相应的批处理文件。

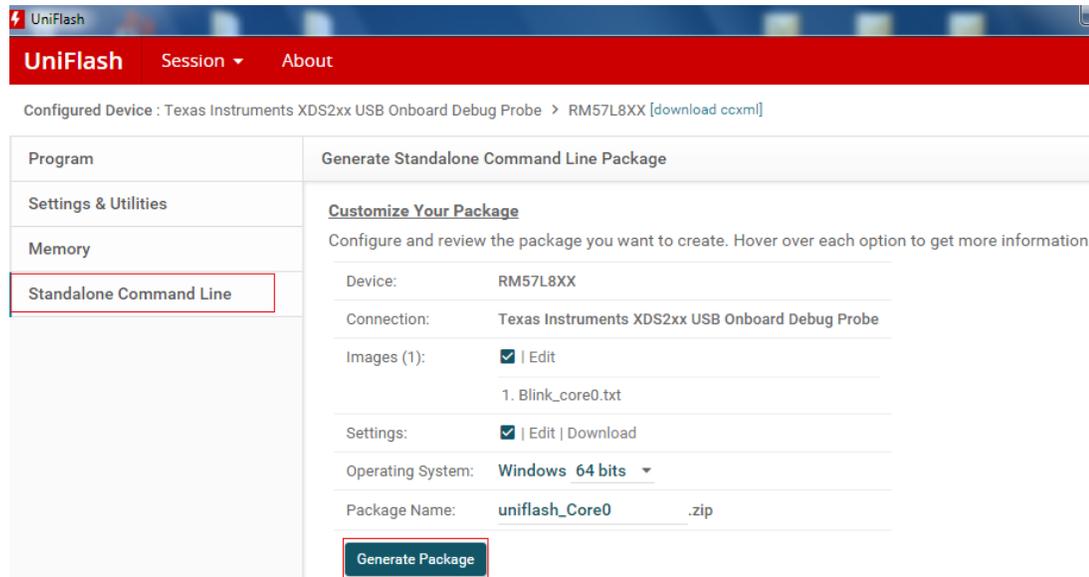


Figure 8. 批处理文件的产生方法

在 Uniflash 的主界面中，点击 Standalone Command Line，会出现 Figure 8 所示界面，用户可根据自己的需要选择下载的固件，修改 Flash 的擦写方式，编程电压，操作系统类型，批处理包的名称等等。配置好后，点击 Generate Package 生成包含批处理文件的压缩包，解压后，在相应 uniflash_Core0 文件夹下，dslite.bat 即是所需要的批处理文件。在运行批处理文件之前，应当先运行 one_time_setup.bat，对系统做一下初始化。针对 Core1 的批处理文件产生方式与 Core0 相同，不再赘述。

3.3.2.2 适用于菊花链电路脚本的修改

在 Uniflash 中，批处理脚本 dslite.bat 是统一生成的，默认下载固件的方式都是针对单核的，并未对 Daisy Chain 作区分。为了兼容 Daisy Chain，需要对 uniflash_Core1/dslite.bat 中 set GENERATED_COMMAND 命令行进行修改，如 Figure 9 方框所示。

```

1  @echo off
2  setlocal
3  setlocal ENABLEDELAYEDEXPANSION
4
5  REM Path to this batch script
6  set UNIFLASH_PATH=%~dp0
7
8  REM Path to DebugServer folder
9  set DEBUGSERVER_ROOT=%~dp0ocs_base\DebugServer\
10
11 set MODE=flash
12 set EXECUTABLE="!DEBUGSERVER_ROOT!bin\DSLite"
13
14 set GENERATED_COMMAND=-c user_files/configs/rm5718xx.ccxml -l user_files/settings/generated.ufsettings -e -f
15
16 REM list available modes
17 if "%1"=="--listMode" (
18     echo.
19
20     echo Usage: dslite --mode ^<mode^> arg
21     echo.

```

Figure 9. dslite.bat 批处理文件脚本

其中，

```
set GENERATED_COMMAND=-c user_files/configs/rm5718xx.ccxml
-l user_files/settings/generated.ufsettings -e -f -v user_files/images/blink_core1.txt
```

需要改写成

```
set GENERATED_COMMAND=-c user_files/configs/rm5718xx.ccxml
-l user_files/settings/generated.ufsettings -n 1 -e -f -v user_files/images/ blink_core1.txt
```

对于 uniflash_Core0/dslite.bat 则不需要作修改。因为如果不加 -n 命令，固件默认会被下载到 Daisy Chain 上的第一个内核 Core0，而 Core1 则不会被 Daisy Chain 所识别。在 Uniflash_Core1/dslite.bat 加了 -n 命令后，就可以将固件下载到想下载的 CPU 了。

3.4 实验结果

脚本修改完成后，双击 `uniflash_Core0/dslite.bat` 执行，经过执行命令，初始化仿真器，擦除 Flash，写入固件等一系列操作后，`Blink_core0.txt` 固件会被下载到 Daisy Chain 上的 `RM57L843_core0` 这个微处理器中。下载完成后，经过 `System Reset`，则可实现固件中设定的功能。

```

DSLite version 7.2.0.2096
Configuring Debugger (may take a few minutes on first launch)...
  Initializing Register Database...
  Initializing: IcePick
  Executing Startup Scripts: IcePick
  Initializing: Dap
  Executing Startup Scripts: Dap
  Initializing: CortexR5
  Executing Startup Scripts: CortexR5
  Initializing: IcePick_0
  Executing Startup Scripts: IcePick_0
  Initializing: Dap_0
  Executing Startup Scripts: Dap_0
  Initializing: CortexR5_0
  Executing Startup Scripts: CortexR5_0
Connecting...
GEL: CortexR5: GEL Output:      Memory Map Setup for Flash @ Address 0x0Loading
Program: user_files/images/Blink_core1.txt
  Preparing ...
GEL: CortexR5: GEL Output:      Memory Map Setup for Flash @ Address 0x0 due to
System Reset
  0 of 1250 at 0x0
Erasing Flash
  Erasing Bank 0, Sector 0
  Erasing Bank 0, Sector 1: 1%
  Erasing Bank 0, Sector 2: 2%

```

Figure 10. Uniflash_Core1 dslite.bat 脚本运行结果

类似的，双击 `uniflash_Core1/dslite.bat`，`Blink_core1.txt` 则会被下载到 `RM57L843_core1` 中。Figure 10 所示为运行 `Uniflash_Core1/dslite.bat` 的执行结果。

显而易见，通过对 `Uniflash` 生成的脚本进行修改后，只需分别点击两个批处理文件即可分别对 Daisy Chain 上的两个微处理器 `RM57L843_core0` 和 `RM57L843_core1` 进行固件下载。能大大提高量产效率，最大限度地避免产线工人误操作。

当然了，用户也可以根据需要将两个脚本合成一个，这样即可真正实现产线固件的一键式烧录。

4 参考文献

1. *JTAG Specification / IEEE 1149 Standard*
2. *SPNS215C-RM57L843 Hercules Microcontroller Based on the ARM Cortex-R Core*
3. <http://processors.wiki.ti.com/index.php/XDS200>

有关 TI 设计信息和资源的重要通知

德州仪器 (TI) 公司提供的技术、应用或其他设计建议、服务或信息，包括但不限于与评估模块有关的参考设计和材料（总称“TI 资源”），旨在帮助设计人员开发整合了 TI 产品的应用；如果您（个人，或如果是代表贵公司，则为贵公司）以任何方式下载、访问或使用了任何特定的 TI 资源，即表示贵方同意仅为该等目标，按照本通知的条款进行使用。

TI 所提供的 TI 资源，并未扩大或以其他方式修改 TI 对 TI 产品的公开适用的质保及质保免责声明；也未导致 TI 承担任何额外的义务或责任。TI 有权对其 TI 资源进行纠正、增强、改进和其他修改。

您理解并同意，在设计应用时应自行实施独立的分析、评价和判断，且应全权负责并确保应用的安全性，以及您的应用（包括应用中使用的 TI 产品）应符合所有适用的法律法规及其他相关要求。您就您的应用声明，您具备制订和实施下列保障措施所需的一切必要专业知识，能够 (1) 预见故障的危险后果，(2) 监视故障及其后果，以及 (3) 降低可能导致危险的故障几率并采取适当措施。您同意，在使用或分发包含 TI 产品的任何应用前，您将彻底测试该等应用和该等应用所用 TI 产品的功能。除特定 TI 资源的公开文档中明确列出的测试外，TI 未进行任何其他测试。

您只有在为开发包含该等 TI 资源所列 TI 产品的应用时，才被授权使用、复制和修改任何相关单项 TI 资源。但并未依据禁止反言原则或其他法律授予您任何 TI 知识产权的任何其他明示或默示的许可，也未授予您 TI 或第三方的任何技术或知识产权的许可，该等产权包括但不限于任何专利权、版权、屏蔽作品权或与使用 TI 产品或服务的任何整合、机器制作、流程相关的其他知识产权。涉及或参考了第三方产品或服务的信息不构成使用此类产品或服务的许可或与其相关的保证或认可。使用 TI 资源可能需要您向第三方获得对该等第三方专利或其他知识产权的许可。

TI 资源系“按原样”提供。TI 兹免除对 TI 资源及其使用作出所有其他明确或默示的保证或陈述，包括但不限于对准确性或完整性、产权保证、无复发故障保证，以及适销性、适合特定用途和不侵犯任何第三方知识产权的任何默认保证。

TI 不负责任何申索，包括但不限于因组合产品所致或与之有关的申索，也不为您辩护或赔偿，即使该等产品组合已列于 TI 资源或其他地方。对因 TI 资源或其使用引起或与之有关的任何实际的、直接的、特殊的、附带的、间接的、惩罚性的、偶发的、从属或惩戒性损害赔偿，不管 TI 是否获悉可能会产生上述损害赔偿，TI 概不负责。

您同意向 TI 及其代表全额赔偿因您不遵守本通知条款和条件而引起的任何损害、费用、损失和/或责任。

本通知适用于 TI 资源。另有其他条款适用于某些类型的材料、TI 产品和服务的使用和采购。这些条款包括但不限于适用于 TI 的半导体产品 (<http://www.ti.com/sc/docs/stdterms.htm>)、[评估模块](http://www.ti.com/sc/docs/sampters.htm)和样品 (<http://www.ti.com/sc/docs/sampters.htm>) 的标准条款。

邮寄地址：上海市浦东新区世纪大道 1568 号中建大厦 32 楼，邮政编码：200122
Copyright © 2017 德州仪器半导体技术（上海）有限公司