

LS1043ARDB-如何在 NAND 闪存中部署 TF-A 二进制文件

原文链接: <https://community.nxp.com/docs/DOC-343572>

Cortex-A 的受信任固件 (TF-A) 是 EL3 安全固件的一种实现。TF-A 代替 PPA 担任安全固件角色。

请注意, 本主题中列出的步骤只能在 LSDK 18.12 和更高版本中执行。

要从以前的引导流程 (使用 PPA) 迁移到 TF-A 引导流程, 您需要编译 TF-A 二进制文件 bl2_<boot_mode>.pbl 和 fip.bin, 并在特定的引导介质上刷新这些二进制文件。

对于 NAND 引导, 您需要编译以下 TF-A 二进制文件。

TF-A 二进制名	组成
bl2_nand	<ul style="list-style-type: none">•BL2 二进制文件: 平台初始化二进制文件•用于 NAND 引导的 RCW 二进制文件
fip.bin	<ul style="list-style-type: none">•BL31: 安全的运行时固件•BL32: 受信任的操作系统, 例如 OPTEE (可选)•BL33: U-Boot / UEFI 图像

请按照以下步骤在 NAND 闪存上编译和部署 TF-A 二进制文件 (bl2_nand.pbl 和 fip.bin)。

- 1.从 RCW 源文件编译 PBL 二进制文件
- 2.编译 U-Boot 二进制文件
- 3.[可选]编译 OPTEE 二进制文件
- 4.编译 TF-A 二进制文件 (bl2_nand.pbl 和 fip.bin) 以进行 NAND 引导
- 5.将 TF-A 二进制文件编程到 NAND 闪存

步骤 1: 从 RCW 源文件编译 PBL 二进制文件

您需要编译 rcw_1600_nandboot.bin 二进制文件以构建 bl2_nand.pbl 二进制文件。

克隆“rcw”存储库并“编译”PBL 二进制文件。

- 1.1.\$ git clone https://source.codeaurora.org/external/qoriq/qoriq-components/rcw
- 2.2.\$ cd rcw
- 3.3.\$ git checkout -b <新分支名称> <LSDK 标签>。 例如, \$ git checkout -b LSDK-19.03 LSDK-19.03
- 4.4.\$ cd ls1043ardb

5.如果需要, 请对 rcw 文件进行更改。

可在 rcw / ls1043ardb / RR_FQPP_1455 / 获得 rcw_1600_nandboot.bin, LS1043ARDB 上用于 NAND 引导已编译 PB。

有关包含 RCW 源文件和二进制文件的目录的命名约定的说明, 请参见 rcw / ls1043ardb / README 文件。

步骤 2: 编译 U-Boot 二进制文件

您需要编译 u-boot.bin 二进制文件以构建 fip.bin 二进制文件。

克隆 u-boot 存储库，并为 TF-A 编译 U-Boot 二进制文件。

```
1.$ git clone https://source.codeaurora.org/external/qoriq/qoriq-components/u-boot.git
2.$ cd u-boot
3.$ git checkout -b <新分支名称> LSDK- <LSDK 版本>。 例如， $ git checkout -b LSDK-19.03 LSDK-19.03
4.$ export ARCH = arm64
5.$ export CROSS_COMPILE = aarch64-linux-gnu-
6.$
7.$ ls1043ardb_tfa_defconfig
8.$
```

如果 make 命令显示错误“***Your GCC is older than 6.0 and is not supported”，请确保使用 Ubuntu 18.04 64 位版本来构建 LSDK 18.12 U-Boot 二进制文件。

编译后的 U-Boot 二进制文件 u-boot.bin 可从 u-boot / 获得。

步骤 3: [可选]编译 OPTEE 二进制文件

您需要编译 tee.bin 二进制文件以使用 OPTEE 构建 fip.bin。但是，OPTEE 是可选的，如果要构建不带 OPTEE 的 FIP 二进制文件，则可以跳过编译 OPTEE 的过程。

克隆 optee_os 存储库并构建 OPTEE 二进制文件。

```
1.$ git clone https://source.codeaurora.org/external/qoriq/qoriq-components/optee_os
2.$ cd optee_os
3.$ git checkout -b <新分支名称> LSDK- <LSDK 版本>。 例如， $ git checkout -b LSDK-19.03 LSDK-19.03
4.$ export ARCH =
5.$ export CROSS_COMPILE = aarch64-linux-gnu-
6.$ make CFG_ARM64_core = y PLATFORM = ls-ls1043ardb
7.$ aarch64-linux-gnu-objcopy -v -O out / arm-plat-ls / core / tee.elf out / arm-plat-ls / core / tee.bin
```

编译的 OPTEE 映像 tee.bin 可从 optee_os / out / arm-plat-ls / core / 获得。

步骤 4: 编译 TF-A 二进制文件以进行 NAND 引导

克隆 atf 存储库，并编译 TF-A 二进制文件 bl2_nand.pbl 和 fip.bin。

```
1. 1.$ git clone https://source.codeaurora.org/external/qoriq/qoriq-components/atf
2. 2.$ cd atf
3. 3.$ git checkout -b <新分支名称> LSDK- <LSDK 版本>。 例如， $ git checkout -b LSDK-19.03 LSDK-19.03
4. 4.$ export ARCH = arm64
5. 5.$ export CROSS_COMPILE = aarch64-linux-gnu-
```

用 OPTEE 构建 BL2 二进制文件。
\$ make PLAT = ls1043ardb bl2 SPD = opteded BOOT_MODE = nand BL32 = <path_to_optee_binary> /tee.bin pbl

RCW = <path_to_rcw_binary> /rcw_1600_nandboot.bin

编译的 BL2 映像 bl2.bin 和 bl2_nand.pbl 可从 atf / build / ls1043ardb / release / 获得。

对于 BL2 源代码或 RCW 二进制文件中的任何更新，都需要重新编译 bl2_nand.pbl 二进制文件。

要在没有 OPTEE 的情况下编译 BL2 二进制文件:

```
$make PLAT=ls1043ardb fip bl2 BOOT_MODE=nand pbl  
RCW=<path_to_rcw_binary>/rcw_1600_nandboot.bin
```

6. 使用 OPTEE 构建 FIP 二进制文件, 而无需受信任的板引导。

```
$ make PLAT = ls1043ardb fip BL33 = <path_to_u-boot_binary> /u-boot.bin SPD = opteed  
BL32 = <path_to_optee_binary> /tee.bin
```

编译的 BL31 和 FIP 二进制文件 bl31.bin, fip.bin 可从 atf / build / ls1043ardb / release / 获得。

对于 BL31, BL32 或 BL33 二进制文件中的任何更新, 都需要重新编译 fip.bin 二进制文件。

要在没有 OPTEE 和没有受信任的板引导的情况下编译 FIP 二进制文件, 请执行以下操作:

```
$ make PLAT = ls1043ardb fip BOOT_MODE = nand BL33 = <_u-boot_binary> /u-boot.bin
```

要使用受信任的板引导程序编译 FIP 二进制文件, 请参阅 <atf> /plat/nxp/README.TRUSTED_BOOT 上的自述文件。

步骤 5: 将 TF-A 二进制文件编程为 NAND 闪存

1. 从 NOR 闪存启动 LS1043ARDB。确保将开关设置为从 NOR bank 0 引导板。要从 NOR bank 0 引导, 开关设置如下:

```
SW3 [1: 8] = 10110011
```

```
SW4 [1: 8] = 00010010
```

```
SW5 [1: 8] = 10100010
```

2. 从 NOR bank 0 引导: => cpld reset

对于 LS1043ARDB, 在启动日志中, 您将看到:

开发板: LS1043ARDB, 从 vBank 0 引导

设置以太网连接

板启动时, U-Boot 将打印已启用的以太网接口的列表。

FM1 @ DTSEC1, FM1 @ DTSEC2, FM1 @ DTSEC3 [PRIME], FM1 @ DTSEC4, FM1 @ DTSEC5

1. 将服务器 IP 地址设置为配置了 TFTP 服务器的主机的 IP 地址。

```
=> setenv serverip <IP 地址 1>
```

2. 将 ethact 和 ethprime 设置为连接到 TFTP 服务器的以太网接口。

请参阅 LS1043ARDB 以太网和 FMC 端口映射, 以获取出现在机箱前面板上的以太网端口名称与 U-Boot 和 Linux 中端口名称的映射。

```
=> setenv ethprime <name of interface connected to TFTP  
server>
```

例如:

```
=> setenv ethprime FM1@DTSEC4
```

```
=> setenv ethact <name of interface connected to TFTP  
server>
```

例如:

```
=> setenv ethact FM1@DTSEC4
```

3. 置单板的 IP 地址。您可以设置静态 IP 地址, 或者, 如果板卡可以连接到 dhcp 服务器, 则可以使用 dhcp 命令。

静态 IP 地址分配:

```
=> setenv ipaddr <ipaddress2>
```

```
=> setenv 网络掩码<子网掩码>
```

动态 IP 地址分配:

```
=> DHCP
```

保存设置。 => saveenv

5.检查单板与 TFTP 服务器之间的连接。

```
=> ping $ serverip
```

```
FM1 @ DTSEC4 device
```

```
192.168.1.1
```

从 TFTP 服务器加载 TF-A 二进制文件

有关 TF-A 二进制文件的闪存映像布局的详细信息，请参阅 TFDK 引导流程的 LSDK 内存布局。

1.Flash bl2_nand.pbl:

```
=> tftp 82000000 bl2_nand.pbl
```

```
=> nand 0x0 $ filesize; nand 0x82000000 0x0 $ filesize;
```

2.Flash fip.bin:

```
=> tftp 82000000 fip.bin
```

```
=> nand erase 0x100000 $ filesize; nand 0x82000000 0x100000 $ filesize;
```

3.从 NAND 闪存引导: => cpld reset nand

LS1043ARDB 将用 TF-A 引导。 在启动日志中，您将看到:

```
Fixed DDR on board
```

```
NOTICE: 2 GB DDR4, 32-bit, CL=11, ECC off
```

```
NOTICE: BL2: v1.5(release):LSDK-19.03
```

```
NOTICE: BL2: Built : 14:46:39, Jun 13 2019
```

```
NOTICE: BL31: v1.5(release):LSDK-19.03
```

```
NOTICE: BL31: Built : 14:52:37, Jun 13 2019
```

```
NOTICE: Welcome to LS1043 BL31 Phase
```

U-Boot 2018.09 (Jun 13 2019 - 12:27:15 +0530)
SoC: LS1043AE Rev1.1 (0x87920011)
Clock Configuration:
CPU0(A53):1600 MHz CPU1(A53):1600 MHz CPU2(A53):1600 MHz
CPU3(A53):1600 MHz
Bus: 400 MHz DDR: 1600 MT/s FMAN: 500 MHz
Reset Configuration Word (RCW):
00000000: 08100010 0a000000 00000000 00000000
00000010: 14550002 80004012 e0106000 c1002000
00000020: 00000000 00000000 00000000 00038800
00000030: 00000000 00001100 00000096 00000001
Model: LS1043A RDB Board
Board: LS1043ARDB, boot from NAND