



Rio Chan and Keerthy J

摘要

本应用报告描述了在 DRA821/J7200/J7VCL 平台上使用 Linux 实现快速引导的过程。此处采集的细节旨在复制 DRA821 上的结果。而此概念可以很轻松地扩展到 Jacinto 7 系列中的其他器件。

本文档使用优化的文件系统 (tisdk-tiny-image) 和减少的 DTSI 在大约 3 秒内实现 Linux 提示符。

内容

1 硬件和软件所需的東西.....	2
2 采用快速引导方法的 DRA821U 引导模式.....	2
3 详细步骤.....	2
3.1 第 1 步：应用补丁并构建/复制 u-boot.....	3
3.2 步骤 2：通过切换到 xSPI 引导进行优化：将引导加载程序复制到 xSPI.....	4
3.3 步骤 3：通过禁用对 Linux 引导来说不是必需的节点来优化 DTSI.....	4
3.4 步骤 4：创建可引导 SD 卡，切换到 TinyFS.....	6
3.5 步骤 5：切换到 eMMC 文件系统.....	6
3.6 步骤 6：使用引导参数进行优化.....	7
3.7 步骤 7：控制 init.....	7
4 调试命令.....	8
4.1 SF 探针.....	8
4.2 mmcblk.....	8
4.3 如何检查已安装的器件？.....	9
4.4 如何检查您的分区？.....	9
4.5 如何恢复您的引导设置？.....	10
5 快速引导结果审查.....	10
6 参考文献.....	10
7 修订历史记录.....	11

插图清单

图 2-1. Jacinto DRA821U Linux 引导流程.....	2
图 4-1. uboot 中的 Jacinto DRA821U SF 探针.....	8
图 4-2. Jacinto DRA821U mmcblk.....	8
图 4-3. Jacinto DRA821U 检查已安装的器件.....	9
图 4-4. Jacinto DRA821U 检查分区.....	9
图 5-1. Jacinto DRA821U 快速引导结果.....	10

表格清单

表 3-1. Jacinto DRA821U EVM 引导开关设置.....	3
--	---

商标

所有商标均为其各自所有者的财产。

1 硬件和软件所需的東西

- 硬件：
 - TI DRA821U EVM
- 软件
 - J7200XSOMXEVM
 - Linux SDK 版本：ti-processor-sdk-linux-j7200-evm-08_00_00_05
 - RTOS SDK 版本：ti-processor-sdk-rtos-j7200-evm-08_00_00_12

关于如何为 DRA821U 构建 Linux/RTOS，请参阅了解 [Processor SDK J7200 \(RTOS/Linux\)](#) 中的引导加载程序章节。

关于如何设置 EVM 进行测试，请参阅 [Jacinto7 DRA821 评估模块 \(EVM\) 用户指南](#)。

2 采用快速引导方法的 DRA821U 引导模式

图 2-1 显示当前使用 SDK 的 Linux 引导流程。

SPL: Linux boot

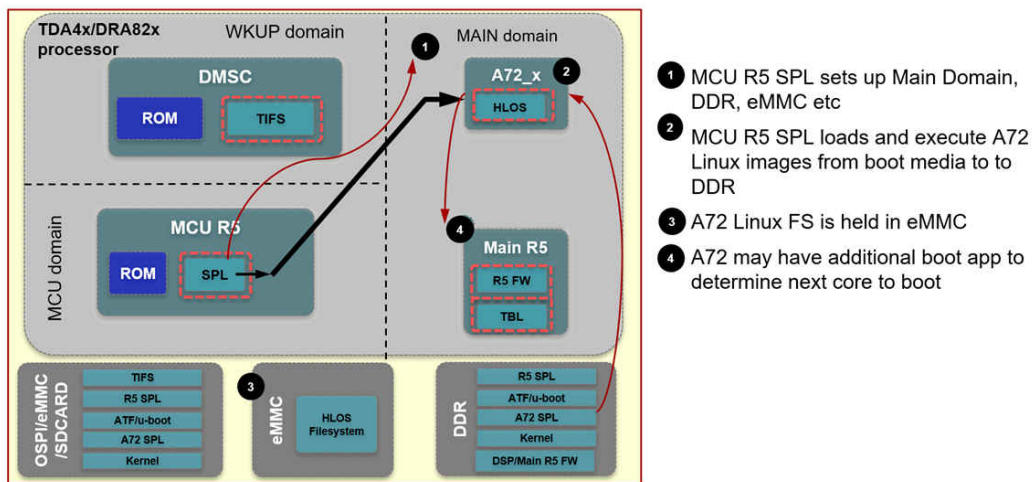


图 2-1. Jacinto DRA821U Linux 引导流程

R5 SPL --> A72 ATF --> A72 SPL --> A72 U-Boot --> A72 Linux。

在上述引导流程中，每个阶段都经过优化，以实现更短的引导时间：

- 优化 U-Boot 以删除 UART 打印。
- 仅启用引导 Linux 所必需的外设，禁用其余外设。
- 切换到从 xSPI 引导，而不是从 SD 卡引导。
- 切换到 eMMC 中的文件系统，而不是使用 SD 卡托管文件系统。

引导到 Linux 内核所花费的时间约为 3 秒。

3 详细步骤

该过程细分为七个步骤。请严格按照这些步骤操作。

- 步骤 1：应用补丁并构建/复制 u-boot 到 SD 卡“引导分区”上。
- 步骤 2：通过切换到 xSPI 引导来优化引导时间，所以我们将引导加载程序从 SD 卡复制到 xSPI 闪存上。
- 步骤 3：通过禁用对 Linux 引导来说不是必需的节点来优化器件树。
- 步骤 4：创建可引导 SD 卡，然后切换到 TinyFS。
- 步骤 5：切换到 eMMC TinyFS 文件系统（非常重要的一步！！）

- 步骤 6：在 uboot 中使用引导参数进行优化。
- 步骤 7：控制 init。

理解这一点：

- 当使用从未被格式化的全新 SD 卡时：请遵循以下顺序：
 - 步骤 4 > 步骤 1 > 步骤 2 > 步骤 3 > 步骤 5 > 步骤 6 > 步骤 7
- 如果使用 DRA821 可引导 SD 卡，请遵循以下顺序：
 - 步骤 1 > 步骤 2 > 步骤 3 > 步骤 4 > 步骤 5 > 步骤 6 > 步骤 7

关于 DRA821U EVM 上的引导开关跳线设置，请参阅表 3-1。

表 3-1. Jacinto DRA821U EVM 引导开关设置

WKUP 引导模式	2	3	4	5	6	7	8	9
DIP SW9	(SW9.1)	(SW9.2)	(SW9.3)	(SW9.4)	(SW9.5)	(SW9.6)	(SW9.7)	(SW9.8)
SD 引导 (默认)	关闭	关闭	关闭	关闭	关闭	关闭	关闭	关闭
eMMC	关闭	打开	关闭	关闭	关闭	关闭	关闭	关闭
OSPI	关闭	打开	关闭	关闭	关闭	关闭	关闭	关闭
UART	关闭	打开	打开	打开	关闭	关闭	关闭	关闭
USB	关闭	关闭	打开	关闭	关闭	关闭	关闭	关闭
无引导	关闭	打开	打开	打开	关闭	关闭	关闭	关闭

主引导模式	0	1	2	3	4	5	6	7
DIP SW8	(SW8.1)	(SW8.2)	(SW8.3)	(SW8.4)	(SW8.5)	(SW8.6)	(SW8.7)	(SW8.8)
SD 引导 (默认)	打开	关闭	关闭	关闭	关闭	关闭	打开	关闭
eMMC	打开	关闭	关闭	关闭	关闭	关闭	关闭	关闭
OSPI	关闭	关闭	关闭	关闭	关闭	打开	打开	关闭
UART	关闭	关闭	关闭	关闭	关闭	关闭	关闭	关闭
USB	打开	关闭	关闭	关闭	关闭	关闭	关闭	关闭
无引导	打开	关闭	关闭	关闭	打开	关闭	关闭	关闭

如上所示，每一步都需要正确匹配引导开关。本文档总结了两种引导开关设置：一种是 SD 卡开关，另一种是 xSPI 开关。

在步骤 2 中：

- 使用待引导的 SD 卡
- 执行一些将 uboot 复制到 xSPI 闪存上的操作。
- 切换到 xSPI 闪存。

应标记每一步中的开关设置。

3.1 第 1 步：应用补丁并构建/复制 u-boot

假设您熟悉 TI Jacinto SDK 构建设置。确保您的 NB 在构建时连接到网络。因为 uboot 可能会引用 ti github。工作副本命令是这样的，您需要根据红色参数 (root) 进行更改。

引导开关设置：

此步骤无需触摸 EVM 上的引导开关。

补丁：

- 01 - Remove-prints-to-optimize-U-Boot-time.patch (该补丁在此处：<https://e2e.ti.com/support/processors-group/processors/f/processors-forum/1105700/dra821u-application-note-how-to-linux-fast-boot-on-dra821u-putting-the-required-2-patches-here>)

命令：(在 **ubuntu** 而不是 **EVM** 上执行此操作)

```
cd $PSDKLA/board-support/u-boot-2021.01+gitAUTOINC+53e79d0e89-g53e79d0e89/
git am 0001-Remove-prints-to-optimize-U-Boot-time.patch
cd ../../..
生成 u-boot
```

cp board-support/k3-image-gen-2021.05/tiboot3.bin board-support/u-boot_build/a72/tispl.bin board-support/u-boot_build/a72/u-boot.img /media/root/boot/

3.2 步骤 2：通过切换到 xSPI 引导进行优化：将引导加载程序复制到 xSPI

这一步是将 SD 卡 u-boot 文件复制到 xSPI 上。

所以，此步骤 2 中有 2 次引导开关设置。

先使用 SD 卡进行引导，然后使用 xSPI 开关。

第 1 次引导开关设置：

#EVM 设置 **SD 引导**模式，请参阅 /psdk_rtos_auto/docs/user_guide/J7_EVM_SETUP.html

SW8[1-8] = 1000 0010

SW9[1-8] = 0000 0000 <--Rio: 阅读 **EVM 用户指南**

SW3[1-10] = 0110001001

命令：

在 u-boot 提示下，执行这些操作。(在 **EVM** 上执行此操作！)

sf probe

fatload mmc 1 \${loadaddr} tiboot3.bin; sf update \$loadaddr 0x0 \$filesize;

fatload mmc 1 \${loadaddr} tispl.bin; sf update \$loadaddr 0x100000 \$filesize;

fatload mmc 1 \${loadaddr} u-boot.img; sf update \$loadaddr 0x300000 \$filesize;

完成上述操作后，执行以下操作：

1. 关闭电路板。
2. 将 DIP 开关设置更改为 xSPI，如下所示：

第 2 次引导开关设置：

#EVM 设置 **xSPI 引导**模式

SW8[1-8]: 1000 0010

SW9[1-8]: 0011 0000

SW3[1-10]: 0111 0010 10

在此之后，请勿更改 SW8/SW9/SW3。

之后，您只能使用此设置，无需再次更改。

3.3 步骤 3：通过禁用对 Linux 引导来说不是必需的节点来优化 DTSI

引导开关设置：

此步骤中无需更改引导开关。

将 SD 卡插入主机 PC

补丁：

0001-arch-arm64-boot-dts-ti-k3-j7200-Optimize-DT-for-earl.patch

该补丁在此处：<https://e2e.ti.com/support/processors-group/processors/f/processors-forum/1105700/dra821u-application-note-how-to-linux-fast-boot-on-dra821u-putting-the-required-2-patches-here>

命令：（在 **ubuntu** 而不是 **EVM** 上执行此操作）

```
gedit $PSDKLA/Rules.make
```

更改此行：DESTDIR=/media/\$USER/rootfs

将 SD 卡插入主机 P

```
cd $PSDKLA/board-support/linux-5.10.41+gitAUTOINC+4c2eade9f7-g4c2eade9f7/  
git am 0001-arch-arm64-boot-dts-ti-k3-j7200-Optimize-DT-for-earl.patch  
cd ../../  
make linux  
sudo make linux_install
```

3.4 步骤 4：创建可引导 SD 卡，切换到 TinyFS

引导开关设置：

此步骤中无需更改引导开关：

命令：（在 **ubuntu** 而不是 **EVM** 上执行此操作）

打开 [\\$PSDKLA/bin/mksdboot.sh](#)（这是为了切换到 **TinyFS**）

执行以下更改：

```
root_fs="$sdkdir/filesystem/tisdk-default-image-j7200-evm.tar.xz"
```

到

```
root_fs="$sdkdir/filesystem/tisdk-tiny-image-j7200-evm.tar.xz"
```

执行以下操作：

[sudo dpkg-reconfigure dash](#)（这是非常重要的一步！）

```
sudo ./mksdboot.sh --device /dev/sdb --sdk /opt/ti-processor-sdk-linux-j7200-evm-08_00_00_05/
```

或

```
sudo ./mksdboot.sh --device /dev/sdc --sdk /opt/ti-processor-sdk-linux-j7200-evm-08_00_00_05/
```

将 SD 卡重新插到 [ubuntu](#) 上。

```
make linux
```

```
sudo make linux_install
```

转至 **TargetNFS**，找到 **lib + usr/lib + sbin**，执行这 3 个复制操作，然后执行“同步”。

执行以下操作：

```
sudo cp -r lib/* /media/root/rootfs/lib      (请查看“root”，您需要根据您的 ubuntu 环境进行更改)  
sudo cp -r usr/lib/* /media/root/rootfs/usr/lib  
cp sbin/mkfs.ext4 /media/root/rootfs/mnt/   (这非常重要!!)  
同步 [sync]
```

3.5 步骤 5：切换到 eMMC 文件系统

引导开关设置：

#EVM 设置 **xSPI** 引导模式

SW8[1-8]: 1000 0010

SW9[1-8]: 0011 0000

SW3[1-10]: 0111 0010 10

命令：(在 EVM 上执行此操作)

将 SD 卡插到 EVM 上。

在 **uboot** 提示符下执行此操作

- gpt write mmc 0 \${partitions}

在内核控制台中执行此操作

- mkfs.ext4 /dev/mmcblk0p1
- mount /dev/mmcblk0p1 /mnt/emmc
- mount /dev/mmcblk1p2 /mnt/sd
- cp -r /mnt/sd/* /mnt/emmc/
- 同步[sync]

3.6 步骤 6：使用引导参数进行优化

引导开关设置：

#EVM 设置 **xSPI** 引导模式

SW8[1-8]: 1000 0010

SW9[1-8]: 0011 0000

SW3[1-10]: 0111 0010 10

命令：(在 EVM 上执行此操作)

注意！

您可以看到这里有 **mmcblk0p1**，因此，您需要通过在内核中发出“fdisk -l”来检查，以确保您拥有这个分区。

在 **uboot** 提示符下，执行以下操作：

```
setenv bootdelay 0
setenv mmcdev 0
setenv bootpart 0
setenv args_mmc "run finduuid;setenv bootargs console=ttyS2,115200n8 root=/dev/mmcblk0p1 rw
rootfstype=ext4 rootwait loglevel=0"
saveenv
boot
```

3.7 步骤 7：控制 init

引导开关设置：

#EVM 设置 **xSPI** 引导模式

SW8[1-8]: 1000 0010

SW9[1-8]: 0011 0000

SW3[1-10]: 0111 0010 10

命令：(在 EVM 上执行此操作)

一旦您进入 Linux shell 提示符。使用以下内容创建文件 `/home/root/init.sh`：

- `#!/bin/sh`
- `mount -t proc proc /proc`
- `mount -n -t sysfs none /sys`
- `mount -n -t tmpfs none /run`
- `echo "entering init script"`

- /bin/sh

然后执行以下操作，这称为“控制”。

- chmod +x init.sh
- cd /sbin/
- rm init
- ln -s /home/root/init.sh init

4 调试命令

以下是一些基本的命令集，用于帮助您调试分区/安装器件。

4.1 SF 探针

当发出 SF 探针时，您将发送此类结果。

您可以看到这三个 uboot 文件正刷写到 xSPI 闪存上。

命令：

Sf probe

```
serial@2800000
serial@2800000
serial@2800000
am65_cpsw_nuss_slave ethernet@46000000: K3 CPSW: nuss_ver: 0x6BA02102 cpsw_ver: 0x6BA82102 ale_ver: 0x00293904 Ports:1 mdio_freq:1000000
Hit any key to stop autoboot: 0
F> sf probe
cadence_spi spi@47040000: Can't get reset: -2
jedec_spi_nor flash@0: non-uniform erase sector maps are not supported yet.
k3-navss-ringacc ringacc@2b800000: Ring Accelerator probed rings:286, gp-rings[96,32] sci-dev-id:235
k3-navss-ringacc ringacc@2b800000: dma-ring-reset-quirk: disabled
SF: Detected s28hs512t with page size 256 Bytes, erase size 256 KiB, total 64 MiB
F> fatload mmc 1 ${loadaddr} uboot3.bin; sf update $loadaddr 0x0 $filesize;
626638 bytes read in 7 ms (71.7 MiB/s)
device 0 offset 0x0, size 0x8092e
626638 bytes written, 0 bytes skipped in 4.60s, speed 132728 B/s
F> fatload mmc 1 ${loadaddr} tisp1.bin; sf update $loadaddr 0x100000 $filesize;
840932 bytes read in 11 ms (72.9 MiB/s)
device 0 offset 0x100000, size 0xcd4e4
840932 bytes written, 0 bytes skipped in 5.593s, speed 153880 B/s
F> fatload mmc 1 ${loadaddr} u-boot.img; sf update $loadaddr 0x300000 $filesize;
1091988 bytes read in 13 ms (80.1 MiB/s)
device 0 offset 0x300000, size 0x10a994
1091988 bytes written, 0 bytes skipped in 7.63s, speed 158250 B/s
F>
```

图 4-1. uboot 中的 Jacinto DRA821U SF 探针

4.2 mmcblk

这是如何检查 mmcblk 的方法

命令：

→ ls /dev/mmcblk*

```
root@j7200-evm:/# ls /dev/mmc* -al
brw----- 1 root root 179, 96 Jan 1 1970 /dev/mmcblk0
brw----- 1 root root 179, 97 Jan 1 1970 /dev/mmcblk0p1
brw----- 1 root root 179, 98 Jan 1 1970 /dev/mmcblk0p2
brw----- 1 root root 179, 0 Jan 1 1970 /dev/mmcblk1
brw----- 1 root root 179, 32 Jan 1 1970 /dev/mmcblk1boot0
brw----- 1 root root 179, 64 Jan 1 1970 /dev/mmcblk1boot1
crw----- 1 root root 237, 0 Jan 1 1970 /dev/mmcblk1rpmb
root@j7200-evm:/#
```

图 4-2. Jacinto DRA821U mmcblk

4.3 如何检查已安装的器件？

命令：

```
cat /proc/mtd
```

```
root@j7200-evm:/# cat /proc/mtd
dev:      size  erasesize  name
mtd0: 00100000 00040000 "ospi.tiboot3"
mtd1: 00200000 00040000 "ospi.tispl"
mtd2: 00400000 00040000 "ospi.u-boot"
mtd3: 00040000 00040000 "ospi.env"
mtd4: 00040000 00040000 "ospi.env.backup"
mtd5: 037c0000 00040000 "ospi.rootfs"
mtd6: 00040000 00040000 "ospi.phypattern"
root@j7200-evm:/#
```

图 4-3. Jacinto DRA821U 检查已安装的器件

4.4 如何检查您的分区？

命令：

```
cat /proc/partitions
```

```
root@j7200-evm:/# cat /proc/partitions
major minor #blocks name
1 0 4096 ram0
1 1 4096 ram1
1 2 4096 ram2
1 3 4096 ram3
1 4 4096 ram4
1 5 4096 ram5
1 6 4096 ram6
1 7 4096 ram7
1 8 4096 ram8
1 9 4096 ram9
1 10 4096 ram10
1 11 4096 ram11
1 12 4096 ram12
1 13 4096 ram13
1 14 4096 ram14
1 15 4096 ram15
31 0 1024 mtdblock0
31 1 2048 mtdblock1
31 2 4096 mtdblock2
31 3 256 mtdblock3
31 4 256 mtdblock4
31 5 57088 mtdblock5
31 6 256 mtdblock6
179 0 15540224 mmcblk0
179 96 60817408 mmcblk1
179 97 63488 mmcblk1p1
179 98 60752896 mmcblk1p2
root@j7200-evm:/#
```

图 4-4. Jacinto DRA821U 检查分区

4.5 如何恢复您的引导设置？

进行引导调试时，如果引导失败，只要您的 SD 卡里面没有损坏，将引导设置切换到 SD 卡，并执行以下命令。您的 DRA821U 将再次可引导。

命令：

- `env default -f -a`
- `setenv board_name j7200`
- `setenv default_device_tree k3-j7200-common-proc-board.dtb`

5 快速引导结果审查

这是在内核登录之前的冷启动，花费时间约为 3 秒。

根据下图中的日志结果，我们得到 $(10.070 - 7.098) = 2.972$ 秒。

对比最初的 Linux 引导（秒数为 2 至 4 倍），这种快速引导结果足以满足汽车应用。

用户可在此基础上进行以下开发：

- 在他们的 DTSI 器件上添加。
- 基于微型 rootfs 在他们的应用上添加。

```

1
2 [0 00:00:07.0985]
3 [0 00:00:07.0985] U-Boot SPL 2021.01-00001-g3837b8de19-dirty (Oct 15 2021 - 09:41:00 +0800)
4 [0 00:00:08.0047] Trying to boot from SPI
5 [0 00:00:08.0219] NOTICE: BL31: v2.5(release):08.00.00.004-dirty
6 [0 00:00:08.0219] NOTICE: BL31: Built : 07:25:50, Aug 7 2021
7 [0 00:00:08.0328]
8 [0 00:00:08.0328] U-Boot SPL 2021.01-00001-g3837b8de19-dirty (Oct 15 2021 - 09:39:45 +0800)
9 [0 00:00:08.0516] Trying to boot from SPI
10 [0 00:00:08.0547] cadence_spi spi@47040000: Can't get reset: -2
11 [0 00:00:08.0547] jedec_spi_nor flash@0: non-uniform erase sector maps are not supported yet.
12 [0 00:00:09.0110]
13 [0 00:00:09.0110]
14 [0 00:00:09.0110] U-Boot 2021.01-00001-g3837b8de19-dirty (Oct 15 2021 - 09:39:45 +0800)
15 [0 00:00:09.0110]
16 [0 00:00:09.0110] 4 GiB
17 [0 00:00:09.0156] Flash: 0 Bytes
18 [0 00:00:09.0156] MMC: sdhci@4fb0000: 0, sdhci@4fb0000: 1
19 [0 00:00:09.0360] serial@2800000
20 [0 00:00:09.0360] serial@2800000
21 [0 00:00:09.0360] serial@2800000
22 [0 00:00:09.0516] am65_cpsw_nuss_slave ethernet@46000000: K3 CPSW: nuss_ver: 0x6BA82102 cpsw_ver: 0x6BA82102 ale_ver: 0x00293904 Ports:1 mdio_freq:1000000
23 [0 00:00:09.0531]
24 [0 00:00:09.0531] Hit any key to stop autoboot: 0
25 [0 00:00:09.0735] switch to partitions #0, OK
26 [0 00:00:09.0735] mmc0(part 0) is current device
27 [0 00:00:09.0953] SD/MMC found on device 0
28 [0 00:00:09.0953] Failed to load 'boot.scr'
29 [0 00:00:09.0969] ** Unrecognized filesystem type **
30 [0 00:00:10.0031] 19137024 bytes read in 59 ms (309.3 MiB/s)
31 [0 00:00:10.0047] 45699 bytes read in 2 ms (21.8 MiB/s)
32 [0 00:00:10.0047] ## Flattened Device Tree blob at 88000000
33 [0 00:00:10.0047] Booting using the fdt blob at 0x88000000
34 [0 00:00:10.0063] Loading Device Tree to 000000008fef1000, end 000000008ffffffe ... OK
35 [0 00:00:10.0078]
36 [0 00:00:10.0078] Starting kernel ...
37 [0 00:00:10.0078]
38 [0 00:00:10.0688] entering init script
39 [0 00:00:10.0688] /bin/sh: can't access tty: job control turned off
40 [0 00:00:10.0703] / #
    
```

图 5-1. Jacinto DRA821U 快速引导结果

6 参考文献

如需使这个快速引导 Linux 正常工作，下面列出的前四个文档是必读材料。最好阅读以下所有文档。

- 德州仪器 (TI) : [Jacinto7 DRA821 评估模块 \(EVM\) 用户指南](#)
- 德州仪器 (TI) : [DRA821 Jacinto™ 处理器数据表](#)
- [适用于 DRA821U u-boot 的 TI 文档](#)
- [快速引导相关的 E2E](#)
- 德州仪器 (TI) : [J7200 DRA821 处理器器件版本 1.0 德州仪器 \(TI\) 产品系列技术参考手册](#)
- [适用于 DRA821U eMMC 刷写的 TI 文档](#)
- [LDconfig](#)

7 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision * (December 2021) to Revision A (July 2022)	Page
• 更新了整个文档中的表格、图和交叉参考的编号格式。.....	2
• 更新了节 3.1 中的补丁。.....	3
• 更新了节 3.3 中的补丁。.....	4

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2022，德州仪器 (TI) 公司