Application Note

*在 **AM62x** 器件上实现最小平台开发*

Jason Kacines

摘要

AM62x Sitara™ MPU 专为进行低成本 Linux® 开发而构建。AM62x 具有多种外设，适用于许多器件，包括汽车和工业应用中的器件。虽然 AM62x 系列可提供许多可靠的特性，但使用 SoC 系列开发定制平台对许多客户来说一直是个挑战。本应用手册通过包含详细的最小平台配置、使用指南以及使用其他 TI SoC 进行此类开发的步骤，可提高利用 AM62x SoC 开发定制电路板的效率。有关 AM62x 产品的更多信息，请参阅器件特定的数据表和用户指南。

内容

表格清单

商标

Sitara™ is a trademark of Texas Instruments.
所有商标均为其各自所有者的财产。

Copyright © 2023 Texas Instruments Incorporated

1 简介

当开发人员购买定制电路板或使用 TI AM62x 系列 SoC 开发定制电路板时，大量集成现有代码库会导致启动时间从几天到几周不等。如果开发人员想要使用 TI 的 AM62x SoC 创建定制电路板，则必须编辑用于 TI 电路板的设备树和配置文件，以适应新的具有不同外设的独立布局。本应用手册介绍了一种最小配置，旨在简化开发过程并将初始启动过程的时间减少到大约一天或更短。

2 开始最小平台开发

2.1 最小平台的功能

在开始开发完全集成的定制电路板之前，这种最小配置可用于通过一组较少的外设测试初始器件功能。为了向开发人员提供更佳的设计，最小配置最初仅支持 SD 卡启动模式。由于定制电路板上所需的硬件有限且启动器件时启动时间较短，因此选择 SD 卡启动模式作为唯一的启动模式。

除了提供开发起点之外，最小配置还通过减慢和禁用组件的许多高级功能来支持广泛的组件。有所降级的两个重要功能包括 SD 读卡器和 DDR 配置。SD 读卡器被迫使用 3.3V 信号以 25MHz 旧版 SDR 运行，以支持更常见的 SD 卡类别。这种支持减少了与高速或低压功能相关的问题数量，如果电路板不能正确支持这些高级功能，则在尝试启动至内核时，此类功能通常会导致故障。默认情况下，最小配置使用 1GB 的 DDR，但可以减少到 512MB，并实施节 2.2 中概述的其他一些更改。此外，默认的 DDR4 配置速度也已从 1600MTs (800MHz) 降至 600MTs (300MHz)，这有助于减少由于信号完整性问题而导致的内核启动故障。此外，不同类型的 DDR（例如 LPDDR4）需要不同的 DDR 配置，必须在尝试启动器件之前生成该配置。

使用此基本配置，可以验证定制电路板的功能，并逐步扩展以包括全系列外设。节 5 中概述了在此启动过程中调试问题的步骤。本指南是面向 AM62x 开发的，但可扩展到节 6 中所述的各种其他 TI SoC。

2.2 最小平台准备

克隆以下存储库：

- 使用**主**分支的 DAS U-Boot (U_BOOT)
- 使用**主**分支的 TI U-Boot (TI_U_BOOT)
- 使用**主**分支的 Arm Trusted Firmware (TFA)
- 使用**主**分支的开放式可移植可信执行环境 (OPTEE)
- 使用 **ti-linux-firmware** 分支的 TI 固件（TIFS、DM、DSMC）

要以最小配置准备 DAS U-Boot，需要移动、修改或创建一些文件。

1. 配置最小设备树和 DDR 配置

   使用以下命令将最小设备树从 TI U-Boot 移至 DAS U-Boot。

```
$ mv path/to/TI_U_BOOT/arch/arm/dts/k3-am625-base.dts path/to/U_BOOT/arch/arm/dts/
$ mv path/to/TI_U_BOOT/arch/arm/dts/k3-am625-r5-base.dts path/to/U_BOOT/arch/arm/dts/
$ mv path/to/TI_U_BOOT/arch/arm/dts/k3-am625-base-u-boot.dts path/to/U_BOOT/arch/arm/dts/
$ mv path/to/TI_U_BOOT/arch/arm/dts/k3-am62x-sk-ddr4-600MTs.dtsi path/to/U_BOOT/arch/arm/dts/
```

   这允许构建脚本访问最小设备树。但是，如果默认 DDR 配置**不**适用于您的定制电路板，则还必须使用 TI 的 SysConfig 对此进行更新，并使用以下对 k3-am625-r5-base.dts 的更改对此进行更新。

```
- #include "k3-am62x-sk-ddr4-600MTs.dtsi"
+ #include "k3-am62x-{DDR_CONFIG}.dtsi"
```

   通过修改以下节点，确保在 k3-am625-base.dts 中明确说明了 SoC 可访问的可用 DDR RAM 容量。

```
memory@80000000 {
     device_type = "memory";
     /* 1 GB RAM (Reduced for Accessibility)*/
     reg = <0x00000000 0x80000000 0x00000000 0x40000000>;
};
```


English Document: SPRADD1
Copyright © 2023 Texas Instruments Incorporated

务必检查设备树，并确保引脚排列与正在测试的定制电路板匹配。如果需要修改设备树中的任何引脚排列，请参阅 TI 的 SysConfig。

CAUTION

对于使用最小 RAM 容量 (512MB) 的器件，在节 2.3 中生成的 U-Boot 映像可能与诸如 OPTEE 之类的关键元件重叠。为防止这种情况导致启动失败，请在 include/configs/am62x_evm.h 中添加以下定义。此更新可保护带有这些关键元件的存储器区域不被 U-Boot 映像覆盖。

```
#define CFG_PRAM (250 * 1024)
```

2. 设置最小 Defconfig

U-Boot 现在支持使用 defconfig 片段。为了正确访问新的最小设备树，请在 path/to/U_BOOT/board/ti/am62x 下创建以下两个 defconfig 片段。

am62x_base_r5.config

```
CONFIG_DEFAULT_DEVICE_TREE="k3-am625-r5-base"
```

am62x_base_a53.config

```
CONFIG_DEFAULT_DEVICE_TREE="k3-am625-base"
CONFIG_SPL_OF_LIST="k3-am625-base"
CONFIG_OF_LIST="k3-am625-base"
```

为了使启动过程能够访问这一新的设备树，必须修改 am62x 的环境文件。修改 path/to/U_BOOT/board/ti/am62x/am62x.env 中的以下行。

```
- default_device_tree=ti/k3-am625-sk.dtb
+ default_device_tree=ti/k3-am625-base.dtb
```

3. 可选：减少最小 Defconfig 大小

为了减小生成的 U-Boot 映像的大小，可以禁用未使用的驱动器以获得最小配置。这主要适用于 R5 配置，因为它存储在受限的 SRAM 中，而 A53 驱动器存储在 DDR 中。如果需要其中任一缩减，请对在步骤 2 中创建的 defconfig 片段应用以下更改。由于最小设备树不会访问 SK/EVM 使用的驱动器，因此此步骤是可选的，不需要使用最小配置进行启动。

am62x_base_r5.config

```
+ # CONFIG_SYS_MALLOC_LEN is not set
+ # CONFIG_SYS_MALLOC_F_LEN is not set
+ # CONFIG_NR_DRAM_BANKS is not set
+ # CONFIG_SF_DEFAULT_SPEED is not set
+ # CONFIG_SF_DEFAULT_MODE is not set
+ # CONFIG_ENV_SIZE is not set
+ # CONFIG_GPIO is not set
+ # CONFIG_DM_GPIO is not set
+ # CONFIG_SPL_DM_SPI is not set
+ # CONFIG_SPL_DRIVERS_MISC is not set
+ # CONFIG_SPL_SPI_FLASH_SUPPORT is not set
+ # CONFIG_SPL_SPI is not set
+ # CONFIG_I2C is not set
+ # CONFIG_INPUT is not set
+ # CONFIG_NET is not set
+ # CONFIG_BOOTDEV_ETH is not set
+ # CONFIG_SPL_DM_SPI_FLASH is not set
+ # CONFIG_SPL_RAM_SUPPORT is not set
+ # CONFIG_SPL_RAM_DEVICE is not set
+ # CONFIG_SPL_SPI_FLASH_TINY is not set
+ # CONFIG_SPL_SPI_FLASH_SFDP_SUPPORT is not set
+ # CONFIG_SPL_SPI_LOAD is not set
+ # CONFIG_SYS_SPI_U_BOOT_OFFS is not set
+ # CONFIG_SPL_YMODEM_SUPPORT is not set
+ # CONFIG_CMD_ASKENV is not set
+ # CONFIG_CMD_DFU is not set
```


Copyright © 2023 Texas Instruments Incorporated

```
+ # CONFIG_SPL_MULTI_DTB_FIT is not set
+ # CONFIG_SPL_MULTI_DTB_FIT_NO_COMPRESSION is not set
+ # CONFIG_SYS_RELOC_GD_ENV_ADDR is not set
+ # CONFIG_DA8XX_GPIO is not set
+ # CONFIG_SPL_MISC is not set
+ # CONFIG_ESM_K3 is not set
+ # CONFIG_MMC_SDHCI_ADMA is not set
+ # CONFIG_SPL_MMC_SDHCI_ADMA is not set
+ # CONFIG_DM_SPI is not set
+ # CONFIG_DM_SPI_FLASH is not set
+ # CONFIG_SPI is not set
+ # CONFIG_SPI_FLASH is not set
+ # CONFIG_SPI_FLASH_SFDP_SUPPORT is not set
+ # CONFIG_SPI_FLASH_SOFT_RESET is not set
+ # CONFIG_SPI_FLASH_SOFT_RESET_ON_BOOT is not set
+ # CONFIG_SPI_FLASH_SPANSION is not set
+ # CONFIG_SPI_FLASH_S28HX_T is not set
+ # CONFIG_CADENCE_QSPI is not set
+ # CONFIG_PINCTRL is not set
+ # CONFIG_PINCTRL_GENERIC is not set
+ # CONFIG_SPL_PINCTRL is not set
+ # CONFIG_SPL_PINCTRL_GENERIC is not set
+ # CONFIG_PINCTRL_SINGLE is not set
```

am62x_base_a53.config

```
+ # CONFIG_SPL_DM_SPI is not set
+ # CONFIG_SPL_SPI_FLASH_SUPPORT is not set
+ # CONFIG_SPL_SPI is not set
+ # CONFIG_SPL_SYS_REPORT_STACK_F_USAGE is not set
+ # CONFIG_SPL_DM_SPI_FLASH is not set
+ # CONFIG_SPL_SPI_FLASH_TINY is not set
+ # CONFIG_SPL_SPI_FLASH_SFDP_SUPPORT is not set
+ # CONFIG_SPL_SPI_LOAD is not set
+ # CONFIG_SYS_SPI_U_BOOT_OFFS is not set
+ # CONFIG_SPL_YMODEM_SUPPORT is not set
+ # CONFIG_SYS_BOOTM_LEN is not set
+ # CONFIG_CMD_PXE is not set
+ # CONFIG_CMD_DHCP is not set
+ # CONFIG_NET_RANDOM_ETHADDR is not set
+ # CONFIG_DMA_CHANNELS is not set
+ # CONFIG_TI_K3_NAVSS_UDMA is not set
+ # CONFIG_MMC_SDHCI_ADMA is not set
+ # CONFIG_SPL_MMC_SDHCI_ADMA is not set
+ # CONFIG_DM_SPI_FLASH is not set
+ # CONFIG_SPI_FLASH is not set
+ # CONFIG_SPI_FLASH_SFDP_SUPPORT is not set
+ # CONFIG_SPI_FLASH_SOFT_RESET is not set
+ # CONFIG_SPI_FLASH_SOFT_RESET_ON_BOOT is not set
+ # CONFIG_SPI_FLASH_SPANSION is not set
+ # CONFIG_SPI_FLASH_S28HX_T is not set
+ # CONFIG_NET is not set
+ # CONFIG_PHY_TI_DP83867 is not set
+ # CONFIG_PHY_FIXED is not set
+ # CONFIG_TI_AM65_CPSW_NUSS is not set
+ # CONFIG_PHY is not set
+ # CONFIG_SPI is not set
+ # CONFIG_DM_SPI is not set
+ # CONFIG_CADENCE_QSPI is not set
```

4. 设置 Binman

要使用 Binman 生成启动所需的二进制文件，必须创建新的 dtsi 文件并进行修改。复制用于 SK/EVM 的原始 binman 配置以实现最小平台。

```
$ cp path/to/U_BOOT/arch/arm/dts/k3-am625-sk-binman.dtsi path/to/U_BOOT/arch/arm/dts/k3-am625-base-binman.dtsi
```

在 k3-am625-base-u-boot.dtsi 中添加以下行以使用此配置。

```
+ #include "k3-am625-base-binman.dtsi"
```


English Document: SPRADD1
Copyright © 2023 Texas Instruments Incorporated

将以下更改添加到刚刚创建的 k3-am625-base-binman.dtsi 中。

```
- #define SPL_AM625_SK_DTB "spl/dts/k3-am625-sk.dtb"
+ #define SPL_AM625_SK_DTB "spl/dts/k3-am625-base.dtb"
```

（可选）更改文件中的**每个**描述以表示基本配置。

```
- description = "k3-am625-sk";
+ description = "k3-am625-base";
```

现在已经在 U-Boot 中设置了最小配置。转到节 2.3，以生成用于启动 SoC 的二进制文件。

2.3 构建二进制文件

本节介绍如何创建二进制文件以使用最小平台配置来启动 SoC。U-Boot 文档中的 K3 代下提供了有关构建通用二进制文件的详细指南，但使用最小配置要求执行一些额外的步骤。

对于 AM62x 器件，启动 SoC 需要三个二进制文件：

- tiboot3.bin
- tispl.bin
- u-boot.img

由于启动过程同时涉及 32 位内核（唤醒域）和 64 位内核（主域），因此需要 32 位和 64 位交叉编译器。以下步骤演练了如何为每个安全级别生成三个必要的二进制文件。以下是 AM62x 器件的三个安全级别。

- GP：通用版本
- HS-FS：高安全性 - 现场安全型
- HS-SE：高安全性 - 强制安全型

CAUTION

生成的二进制文件**仅**适用于具有相同安全级别的器件。

Binman 用于构建启动至内核所需的二进制文件和设备树。适用于 K3 代的 U-Boot 文档概括介绍了如何创建这些二进制文件，但必须进行一些修改才能实现最小配置。以下是最小平台启动所需的步骤。

设置以下通用环境变量：

表 2-1. 通用环境变量

环境变量	说明
CC32	适用于 ARMv7（ARM 32 位）的交叉编译器，通常为 arm-linux-gnueabihf-
CC64	适用于 ARMv8（ARM 64 位）的交叉编译器，通常为 aarch64-linux-gnu-
LNX_FW_PATH	TI Linux 固件库的路径
TFA_PATH	Arm Trusted Firmware 源代码的路径
OPTEE_PATH	OP-TEE 源代码的路径

以下是设置通用环境变量的示例。

```
$ export CC32=arm-linux-gnueabihf-
$ export CC64=aarch64-linux-gnu-
$ export LNX_FW_PATH=path/to/ti-linux-firmware
$ export TFA_PATH=path/to/trusted-firmware-a
$ export OPTEE_PATH=path/to/optee_os
```

设置以下特定于电路板的环境变量：

表 2-2. 特定于电路板的环境变量

环境变量	说明
UBOOT_CFG_CORTEXR	适用于 Cortex-R 的 Defconfig

Copyright © 2023 Texas Instruments Incorporated

表 2-2. 特定于电路板的环境变量 (continued)

环境变量	说明
UBOOT_CFG_CORTEXA	适用于 Cortex-A 的 Defconfig
TFA_BOARD	用于为 Cortex-A 处理器构建 TF-A 的平台名称
TFA_EXTRA_ARGS	用于构建 TF-A 的任何额外参数
OPTEE_PLATFORM	用于为 Cortex-A 处理器构建 OP-TEE 的平台名称
OPTEE_EXTRA_ARGS	用于构建 OP-TEE 的任何额外参数

请注意，TFA_EXTRA_ARGS 和 OPTEE_EXTRA_ARGS 通常留空。下面是一个设置特定于电路板的环境变量的示例。

```
$ export UBOOT_CFG_CORTEXR="am62x_evm_r5_defconfig am62x_base_r5.config"
$ export UBOOT_CFG_CORTEXA="am62x_evm_a53_defconfig am62x_base_a53.config"
$ export TFA_BOARD=lite
$ export OPTEE_PLATFORM=k3
```

构建 tiboot3.bin

```
$ cd path/to/U_BOOT
$ make $UBOOT_CFG_CORTEXR
$ make CROSS_COMPILE=$CC32 BINMAN_INDIRS=$LNX_FW_PATH
```

构建 tispl.bin 和 u-boot.img

```
$ cd path/to/ATF
$ make CROSS_COMPILE=$CC64 ARCH=aarch64 PLAT=k3 SPD=opteed $TFA_EXTRA_ARGS \
       TARGET_BOARD=$TFA_BOARD

$ cd path/to/OPTEE
$ make CROSS_COMPILE=$CC32 CROSS_COMPILE64=$CC64 CFG_ARM64_core=y $OPTEE_EXTRA_ARGS \
       PLATFORM=$OPTEE_PLATFORM

$ cd path/to/U_BOOT
$ make $UBOOT_CFG_CORTEXA
$ make CROSS_COMPILE=$CC64 BINMAN_INDIRS=$LNX_FW_PATH \
       BL31=$TFA_PATH/build/k3/$TFA_BOARD/release/bl31.bin \
       TEE=$OPTEE_PATH/out/arm-plat-k3/core/tee-raw.bin
```


English Document: SPRADD1
Copyright © 2023 Texas Instruments Incorporated

根据所测试的器件的安全级别，使用以下二进制文件。

表 2-3. 根据器件安全级别生成的二进制文件

安全性	生成的二进制文件
GP	tiboot3-am62x-gp-evm.bin tispl.bin_unsigned u-boot.img_unsigned
HS-FS	tiboot3-am62x-hs-fs-evm.bin tispl.bin u-boot.img
HS-SE	tiboot3-am62x-hs-evm.bin tispl.bin u-boot.img

内核的设备树生成为 u-boot.dtb，每个设备都需要它。

生成的三个二进制文件必须确切地重命名为 tiboot3.bin、tispl.bin 和 u-boot.img。生成的设备树 blob 也必须重命名为 k3-am625-base.dtb。这两个步骤都可以使用以下命令来完成。

```
$ mv tiboot3-am62x-{gp/hs-fs/hs}.bin tiboot3.bin
$ mv tispl.bin{_unsigned} tispl.bin
$ mv u-boot.img{_unsigned} u-boot.img
$ mv u-boot.dtb k3-am625-base.dtb
```

3 部署说明

要使用提供的最小平台配置来启动器件，需要使用 SD 卡来启动器件。首先，用预构建的映像蚀刻 SD 卡，以便轻松对 SD 卡进行分区。

- 访问 PROCESSOR-SDK-AM62X
- 点击“PROCESSOR-SDK-LINUX-AM62X - Processor SDK Linux for AM62X”
- 下载所需的内核映像。建议使用默认映像。

CAUTION

对于使用最小 RAM 容量 (512MB) 的器件，默认内核映像可能无法在使用低容量 RAM 的器件上运行。建议使用 1GB RAM 来运行默认内核映像。要下载瘦 Linux 内核映像，请导航至“CI/CD Snapshot for developers”>“am62xx”>“latest”。

运行以下命令解压缩 SD 卡映像。

```
$ xz -d tisdk-{image-name}-am62xx-evm.wic.xz
```

然后，通过您偏好的方法使用 wic 映像蚀刻 SD 卡。本示例使用 dd 文件传输，SD 卡安装在 /dev/sda 上。

```
$ sudo dd if=tisdk-{image-name}-am62xx-evm.wic of=/dev/sda
```

蚀刻完成后，断开并重新连接 SD 卡。现在，您可以看到器件的两个分区：**boot** 和 **root**。

在 **boot** 分区中，移动三个生成的二进制文件并替换现有的二进制文件（tiboot3.bin、tispl.bin 和 u-boot.img）。

在 **root** 分区中，在 boot/dtb/ti 下添加新的设备树。(k3-am625-base.dtb)

卸载 SD 卡。SD 卡现已准备好使用最小配置启动器件。

3.1 验证最小配置功能

将计算机连接到电路板的主 UART0 上的串行连接。为简单起见，建议您不要在此时为整个器件上电。

Copyright © 2023 Texas Instruments Incorporated

创建与电路板上主 UART0 的串行连接，波特率为 115200。在本例中，picocom 用于建立串行连接。对于 AM62x 器件，需要使用 USB0 来建立此连接。

```
$ sudo picocom -b 115200 /dev/ttyUSB0
```

将 SD 卡插入电路板并为其上电。串行端口在器件启动时打印日志，并最终启动到内核，同时显示类似以下打印内容的登录提示。根据使用的内核映像，输出内容可能并不完全相符。

```
 _____                  _____          _         _
|  _  |___ ___ ___ ___  |  _  |___ ___ |_|___ ___| |_
|     |  _| .'| . | . | |   __|  _| . | | | -_|  _|  _|
|__|__|_| |__,|_  |___| |__|  |_| |___|_| |___|___|_|
              |___|                   |___|

Arago Project am62xx-evm ttyS2

Arago {VERSION} am62xx-evm ttyS2

am62xx-evm login:
```

输入 *root* 作为登录名，以访问器件的根目录和应用程序可能所需的任何命令。

如果器件无法成功启动至内核，请按照节 5 中所述的步骤寻找问题的根源。

如果器件成功启动至内核，则可以使用节 4 中概述的步骤扩展最小配置。

4 扩展最小配置

性能降级的两个主要外设是 SD 读卡器和 DDR 配置。要增强这些组件的功能并测试其完整功能，必须编辑设备树。要增强 SD 读卡器的功能，请在 k3-am625-base.dts 中编辑以下节点。

```
/* SDHCI1 (MMC1) is used for SD/MMC boot mode */
&sdhci1 {
     status = "okay";
     pinctrl-names = "default";
     pinctrl-0 = <&main_mmc1_pins_default>;
     ti,driver-strength-ohm = <50>;
     disable-wp;
     /* Disable 1.8v and High Speed Modes for Accessibility */
     /* Force MMC Legacy @ 25MHz bus speed */
     no-1-8-v;
     sdhci-caps-mask = <0x00000000 0x00200000>;
};
```

要增强 DDR 配置的功能，请访问 TI 的 SysConfig 并生成时钟频率更高的配置。如果使用了原始最小平台 DDR 配置 (k3-am625-sk-ddr4-600MTs)，请将 k3-am625-r5-base.dts 中使用的 DDR 配置更改为 k3-am625-sk-ddr4-1600MTs。

一旦这些元件及其功能得到增强以反映自定义平台，就开始根据需要为器件添加新的设备树节点和驱动器。有关开发设备树的提示，请参阅位于以下文件中的由 TI-AM625-SK/EVM 使用的设备树。

```
k3-am625-sk.dts
k3-am625-r5-sk.dts
k3-am625-sk-u-boot.dtsi
```


English Document: SPRADD1
Copyright © 2023 Texas Instruments Incorporated

5 工具和调试

当尝试启动定制电路板时，预计在整个启动过程中会遇到问题。本节介绍了启动定制电路板时出现的常见问题以及如何处理调试过程。

5.1 常见问题

如果器件无法启动至 U-Boot 提示符或没有向控制台打印任何内容，请确保针对所测试器件的安全级别（GP、HS-FS、HS-SE）使用正确的器件二进制文件。如果 TIFS 固件（例如，HS-FS 器件上的 GP 固件）不匹配，则通常看不到控制台上有任何打印内容，因为 ROM 无法正确读取固件并立即中止。

当使用最小配置时，有两个主要的硬件问题可能导致器件无法启动至 U-Boot 提示符。SD 读卡器或 DDR 的配置或它们与 SoC 的连接可能不正确。确保 DDR 配置与连接到 SoC 的硬件匹配。如果未正确设置 DDR 配置，则常见的故障点是在加载 SPL 期间。如果仅将以下内容打印至控制台，则很可能是 DDR 配置出现问题。

```
U-Boot SPL {version} ({date} - {time})
SYSFW ABI: {version} (firmware rev {revision})
SPL initial stack usage: {size} bytes
```

如果器件在 U-Boot 提示符后出现故障（在内核启动序列期间），则可能是设备树出现问题。请检查 dtb 是否已放置在根文件系统中可供内核访问的正确位置。如果器件仍然无法启动到内核，或者在成功启动到内核提示符后崩溃，建议尝试使用较小的内核映像进行启动。瘦 linux 内核映像应该在任何具有至少 512MB 的 DDR RAM 的 AM62x 器件上启动。有关设备树和使用其他内核映像的更多详细信息，请参阅节 3。

使用默认内核映像时的另一个常见问题是由未禁用的以太网节点导致的。如果在尝试运行以太网驱动器时，器件在启动至内核时失败，请将以下设备树节点添加到 k3-am625-base.dts

```
&cpsw3g {
     status = "disabled";
};
```

5.2 OpenOCD 调试

如果节 5.1 没有解决问题，建议使用 JTAG 调试电路板以找出问题的根源。OpenOCD 是一款开源调试器，可以与各种基于 gdb 的调试 IDE 交互。已为 K3 系列器件创建了 OpenOCD 使用指南，该指南位于 U-Boot 文档中的 K3 代下。

Copyright © 2023 Texas Instruments Incorporated

6 未来的工作

虽然这种最小配置是专门为 AM62x 开发的，但该配置可以扩展到 K3 系列中的其他器件。建议使用这种最小配置作为基础，并在另一个 SoC 中包含等效元件。以下步骤可用作创建最小配置的通用方法。

- 为 U-Boot 和 Linux 内核创建最小设备树
  - 这些设备树需要能实现所需的 DDR 配置并使用 3.3V 信号以 25MHz 读取 SD 卡。

```
k3-{SOC}-base.dts k3-{SOC}-r5-base.dts k3-{SOC}-base-u-boot.dtsi
```

- 创建代表所需元件的最小 defconfig 片段
  - 这些 defconfig 片段不是必需的，如果不使用，只会导致更大的 U-Boot 映像。设备树指定 SoC 将使用其中哪些驱动器，并且只导致最少的存储器浪费量（最多大概为 500KB）。这些片段在现有的 am62x_evm_a53_defconfig 和 am62x_evm_r5_defconfig 基础上构建，以禁用未使用的驱动器。

```
{SOC}_base_a53.config {SOC}_base_r5.config
```

如果使用最小容量的 DDR (512MB)，则务必要了解映像的组织方式。值得注意的是，ATF 和 OPTEE 的位置可以避免 U-Boot 映像覆盖这些必要元件所在的部分。由于每个器件都包含 SoC 本身所需的元件，因此没有详细说明创建新的最小配置的过程。

English Document: SPRADD1
Copyright © 2023 Texas Instruments Incorporated

7 总结

通过减少外设数量并降级所启用外设的功能，可以为 SoC 创建最小配置以作为启动定制电路板的初始测试。本应用手册旨在简化使用 TI SoC 创建定制电路板的过程，其中概述了这一过程并包括该过程的详细步骤。通过采用最小配置，可简化开发过程，并将初始启动过程的时间减少至一天左右。

8 参考文献

- 德州仪器 (TI)，*AM62x 处理器器件版本 1.0 技术参考手册*。
- 德州仪器 (TI)，*适用于 AM62x 的 Processor SDK Linux*。
- *U-Boot 文档*。
- *OpenOCD 用户指南*。
- 所有相关补丁：
  - *U-Boot K3 系列 OpenOCD 指南*。
  - *U-Boot 配置片段更改讨论*。

Copyright © 2023 Texas Instruments Incorporated

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出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 © 2023，德州仪器 (TI) 公司