

## Application Note

**TDA4 : 定制电路板启动指南**

Keerthy J, Karthik R, and Erick Narvaez

**摘要**

TDA4-EVM 是一个参考开发平台，该平台具有可展示片上系统 (SoC) 功能的外设超集。客户根据需要其平台支持的用例开发电路板。通常，减少外设、更改所选的外设器件和改变支持的外设实例数最终会产生不同的设计。因此，除了默认 SDK 之外，客户还需要根据电路板设计，实施与定制引脚多路复用、DDR、PMIC、控制台通用异步接收器/发送器 (UART) 相关的更改。本应用手册介绍了 SDK 基础之上所需的更改，并描述了定制 TDA4 电路板启动过程中遇到的常见问题。

**内容**

<b>1 PMIC 和电源定制更改</b> .....	<b>2</b>
<b>2 Pinmux</b> .....	<b>3</b>
2.1 配置 PinMux 的步骤.....	3
2.2 引脚多路复用的手动审核.....	4
<b>3 定制 DDR 相关更改</b> .....	<b>5</b>
<b>4 开始使用的最小内核 DT</b> .....	<b>5</b>
<b>5 引导模式支持汇总</b> .....	<b>5</b>
5.1 无引导模式.....	5
5.2 UART 引导模式.....	5
5.3 OSPI/QSPI/SPI/xSPI/串行与非门.....	5
<b>6 定制电路板启动期间的常见问题</b> .....	<b>6</b>

**商标**

所有商标均为其各自所有者的财产。

## 1 PMIC 和电源定制更改

定制电路板可以具有不同的 PMIC 来提供 SoC 电压域。可能会由于成本优势、不同的电源要求和可用性而选择不同的 PMIC。PMIC 配置通常在安全型 MCU R5F0 中完成。R5 SPL 主要使用 AVS 补偿电压来配置 CPU 电源轨。需要在 2 个位置对定制 PMIC 进行更改：

1. 器件树：需要使用正确的 PMIC 节点来填充 WKUP\_I2C0 的器件树。示例：arch/arm/dts/k3-j721e-r5-common-proc-board.dts。连接正确的稳压器节点作为 VTM 节点的电源。

```
&wkup_i2c0 {
    u-boot,dm-spl;
    tps659413a: tps659413a@48 {
        reg = <0x48>;
        compatible = "ti,tps659413";
        u-boot,dm-spl;
        pinctrl-names = "default";
        pinctrl-0 = <&wkup_i2c0_pins_default>;
        clock-frequency = <400000>;

        regulators: regulators {
            u-boot,dm-spl;
            buck12_reg: buck12 {
                /*VDD_MPU*/
                regulator-name = "buck12";
                regulator-min-microvolt = <800000>;
                regulator-max-microvolt = <1250000>;
                regulator-always-on;
                regulator-boot-on;
                u-boot,dm-spl;
            };
        };
    };
};

&wkup_vtm0 {
    vdd-supply-2 = <&buck12_reg>;
    u-boot,dm-spl;
};
```

2. configs/j721e\_evm\_r5\_defconfig 中 PMIC 的配置更改。例如，以下是 TPS65941 PMIC 所需的更改：
3.
 

```
CONFIG_DM_PMIC=y
CONFIG_PMIC_TPS65941=y
CONFIG_DM_REGULATOR=y
CONFIG_SPL_DM_REGULATOR=y
CONFIG_DM_REGULATOR_TPS65941=y
```

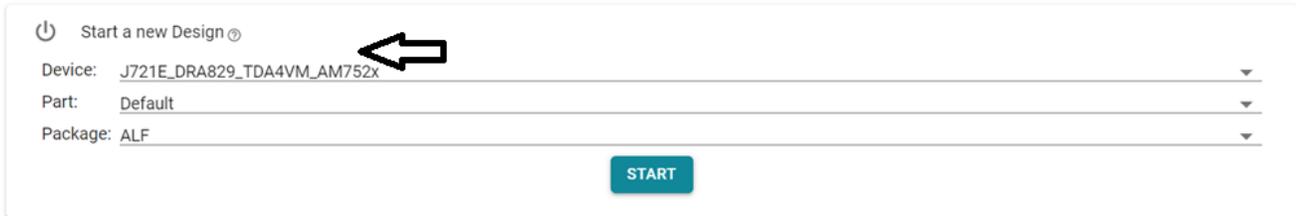
## 2 Pinmux

为 Jacinto 器件配置引脚多路复用的建议方法是使用 **PinMux 工具**。该工具可在线使用，也提供了安装程序供离线使用。

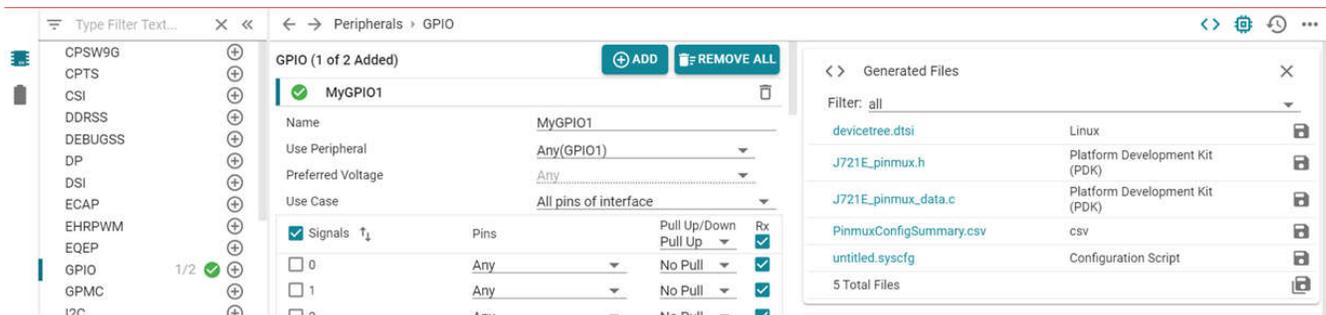
### 2.1 配置 PinMux 的步骤

1. 转到 <https://dev.ti.com/sysconfig/?fromPinmux=true#/start>。
2. 在“Start a new Design”中，选择一个与器件型号相匹配的现有设计。例如，J721E\_DRA829\_TDA4VM\_AM752x。

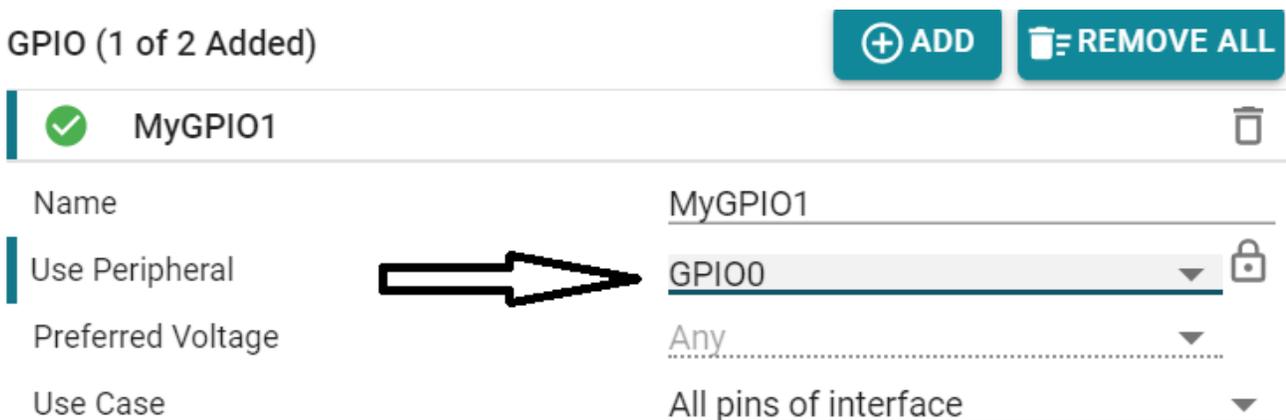
保留“Part”选项卡的默认值，并将“Package”保留为 ALF。点击“Start”按钮，如以下截图所示：



3. 通过在“Type Filter Text”（页面左上角）中输入相应的名称来选择外设。例如，选择 GPIO 并点击“+”号。这将在中心添加一个模块。



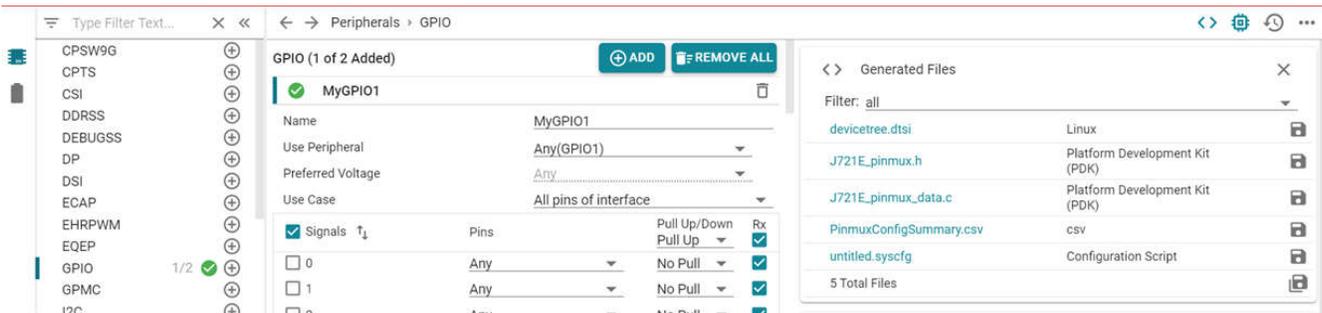
4. MyGPIO1 有一些默认值。在“Use peripheral”菜单中选择确切的模块实例，默认为 ANY。使用下拉菜单选择相应的实例。本例中使用 GPIO0。



- 这样我们就完成了模块级选择。现在，我们只需选择与电路板原理图中的焊球名称对应的焊球名称。这可以通过在“Pins”下拉菜单中选择正确的焊球名称来实现。根据原理图，从下拉菜单中选择相应的“Pull Up/Down”。有关如何选择引脚名称 EXTINTn 的信息，请参阅下面的 **引脚多路复用的手动审核** 详细信息。



- 根据选择的 OS，从“Generated Files”部分下载以下文件之一：- Linux：devicetree.dtsi：只需复制节点并将其粘贴到正确的父引脚多路复用节点之下（有关详细信息，请参阅下文）- PDK(RTOS)：J721E\_pinmux.h/J721E\_pinmux.c。



- 例如，Linux：devicetree.dtsi。

我们选择了主 gpio0，因此在 arch/arm64/boot/dts/ti/j721-common-proc-board.dts 文件中示例 mygpio1\_pins\_default 的引脚多路复用节点应位于 main\_pmx0 节点之下。

```
&main_pmx0 {
    pinctrl-single,pins = < J721E_IOPAD(0x0, PIN_INPUT, 7) /* (AC18) EXTINTn.GPIO0_0 */ >;
};
```

可以对 u-boot 器件树执行相同的操作。我们采用相同的示例文件 arch/arm/dts/j721-common-proc-board.dts。

## 2.2 引脚多路复用的手动审核

- 检查原理图中是否存在确切的焊球编号。例如，J721E/TDA4VM(SoC) à gpio0\_0(Signal\_name)。在相应的数据手册中搜索 TDA4VM/J721E：

<https://www.ti.com/cn/lit/ds/symlink/tda4vm.pdf>

[ts=1593521450358&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTDA4VM](https://www.ti.com/product/TDA4VM)

- 表中给出了第一个搜索实例，其中提供了焊球编号和名称，这正是我们需要在电路板原理图中检查的内容，以了解在电路板上的哪个焊球上输出了 gpio0\_0。例如：AC18 映射到 GPIO0\_0。页面：上文 #1 中共享的文档链接的第 18 页。

B29	emu1	EMU1	0	IO	PU	0	1.8 V/3.3 V	VDDSHV0_MCU	Yes	LVCMOS	TBD	1/1
AC18	extintn	EXTINTn	0	I	OFF	7	1.8 V/3.3 V	VDDSHV2	Yes	I2C OPEN DRAIN	TBD	0/0
		GPIO0_0	7	IO							0	

- 搜索映射到 padconfig 寄存器的 AC18，如下所示。在器件特定数据手册的 PADCONFIG0 寄存器中搜索 AC18，这是需要编程的模式。例如，gpio0\_0 处于模式 7，因此 pinmux 文件应采用模式 7，以便从焊球 AC18 中输出 gpio0\_0 信号。页面：上文 #1 中指向的文档链接的第 138 页。

您可以统计由 PinMux 工具生成的 devicetree.dtsi、手动审核的寄存器和位字段，并进行交叉验证。例如，J721E\_IOPAD(0x0, PIN\_INPUT, 7) 0x0 对应于 PADCONFIG0 寄存器的寄存器偏移 0x0，7 对应于多路复用器模式 7。

### 3 定制 DDR 相关更改

定制电路板可以具有不同的 DDR 器件。可能会由于成本优势、不同的存储器要求和电源设计而选择不同的 DDR 器件。所有与 DDR 相关的配置和初始化都在 R5 SPL 中进行。需要生成定制 DT 文件来解决差异。需要将 arch/arm/dts/k3-j721e-ddr-evm-lp4-4266.dtsi 替换为与所选 DDR 器件相对应的定制 dts 文件。这涉及使用 DDR 配置工具。详细信息可在 [Jacinto7 DDRSS 寄存器配置工具](#) 中获取。

### 4 开始使用的最小内核 DT

第一个主要目标是使用 TDA4 在定制电路板上启动 Linux。从仅涵盖引导介质、控制台 UART、计时器和其他必要 DT 节点的最小器件树开始，始终是更好且更轻松的选择。以下 zip 文件包含用于禁用对引导 Linux 非必需的外设节点的补丁。SDK 中提供的默认器件树是 TDA4 支持的所有外设的超集。因此，需要在 DT 中进行优化，以避免由于定制电路板中的外设不匹配而导致崩溃/挂起。选择最小节点集并使用该节点集引导至 Linux 提示符。宣布客户可以启用与其定制电路板相关的节点。

以下是一组用于 8.0 Linux SDK 的参考 DT 优化补丁：

[https://e2e.ti.com/cfs-file/\\_\\_key/communityserver-discussions-components-files/791/0001\\_2D00\\_arm64\\_2D00\\_dts\\_2D00\\_j721e\\_2D00\\_Enable\\_2D00\\_Minimal\\_2D00\\_DT\\_2D00\\_configuration.patch](https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/791/0001_2D00_arm64_2D00_dts_2D00_j721e_2D00_Enable_2D00_Minimal_2D00_DT_2D00_configuration.patch)

### 5 引导模式支持汇总

#### 5.1 无引导模式

广受支持的硬件启动引导模式。GEL 文件是初始化器件主要方法，它将使 SoC 进入一种状态，在该状态下，内核可以加载程序，并且可以在引导过程中以最可控的方式运行测试。GEL 文件随器件的 RTOS SDK 一起提供，或者当器件目标配置本身可用时，GEL 文件也是 Code Composer Studio 的一部分。Lauterbach 脚本通常也可用。

无引导模式使引导过程变得简单，应在硬件检查后立即使用。对于更复杂的应用，建议迁移到 SPL 和 SBL 等引导解决方案。这些解决方案可以在无引导模式下加载，不过需要定制工作，并且通常需要在调试器中执行更多步骤。

#### 5.2 UART 引导模式

UART 引导模式允许将引导二进制文件加载到已由 ROM 初始化的内存中。它是最简单的引导模式，除了 UART 之外不需要任何其他模块，因此非常有用。可以加载 DM 固件，它将接管并加载所需的其余引导二进制文件。UART 引导模式的缺点是速度：它将采用 ROM 设置的默认速度，这可能会影响大型应用程序的加载。对于初始检查来说，这是一种很好的初始引导模式，但在后期阶段，最好使用其他引导模式。

#### 5.3 OSPI/QSPI/SPI/xSPI/串行与非门

从闪存器件引导是常见的生产引导介质选择之一。SDK 提供了用于刷写介质的工具，但在电路板启动期间，这会增加一个故障点和对工具的依赖。建议使用其他引导模式进行备份测试，以防此引导模式出现任何问题。

## 6 定制电路板启动期间的常见问题

1. **U-Boot 中的定制 dts 文件**：U-Boot 框架要求为 A72 SPL 提供对应于 k3-j721e-common-proc-board.dtsi 的 k3-j721e-common-proc-board-u-boot.dtsi。因此，如果定制电路板具有名为 k3-j721e-custom-board.dts 的 DTS 文件，则应添加 k3-j721e-custom-board-u-boot.dts。这对于 A72 SPL 而言是强制性的。
2. 将控制台切换到不同的 UART。例如，如何将控制台从 MAIN\_UART 切换到 MCU\_UART。

根据引导流程，分为三个阶段：

- R5 SPL
- A72 SPL
- A72 u-boot。

因此您需要在所有级别进行此项更改。除此之外，ATF (Arm Trusted Firmware) 还需要根据需要更改控制台端口。以下是 J7200 上的 dts 更改，也可以对 J721e 进行类似更改：

常见问题解答链接：<https://e2e.ti.com/support/processors/f/791/t/988278>

3. 如何在没有控制台输出时使用 CCS 进行调试？

在 R5 SPL 的第一条指令的开头添加无限循环。

```
diff --git a/arch/arm/cpu/armv7/start.S b/arch/arm/cpu/armv7/start.S
index 4f6327fe3ab..96c4554744c 100644
--- a/arch/arm/cpu/armv7/start.S
+++ b/arch/arm/cpu/armv7/start.S
@@ -37,6 +37,7 @@
#endif

reset:
+   b reset
   /* Allow the board to save important registers */
   b save_boot_params
save_boot_params_ret:
```

然后我们可以使用 CCS 连接到 MCU R5F\_0。将 PC 移至下一条指令并单步执行，以检查崩溃位置。

4. 如何在控制台中针对 j721s2 和 j784s4 更改 ATF 和 OPTEE？

Make CROSS\_COMPILE64=aarch64-none-linux-gnu- CROSS\_COMPILE=arm-none-linux-gnueabi-  
PLATFORM=k3-j784s4 CFG\_ARM64\_core=y **CFG\_CONSOLE\_UART=0x8**

基于定制电路板 UART\_INSTANCE X。

CFG\_CONSOLE\_UART=0x8 将会更改

## 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 © 2024，德州仪器 (TI) 公司