

TDA4VEN 和 AM62P 三屏独立显示方案

Fredy Zhang; Jenney Ni;

EP FAE

摘要

德州仪器 (TI) 的 TDA4VEN 和 AM62P 处理器是面向汽车电子和工业人机界面 (HMI) 应用的高集成度 SoC。TDA4VEN 主打高性能计算与图形处理，集成 C7x DSP 和深度学习加速器，适用于需要复杂图形渲染和 AI 处理的数字仪表盘与信息娱乐系统。AM62P 则主打高性价比和低功耗，基于高效的 Arm® Cortex®-A53 内核，适合成本敏感的工业 HMI 和基础型车载显示应用。两款芯片均内置了强大的显示子系统 (Display Sub-System, DSS)，能够高效地驱动多个显示屏，实现丰富的视觉交互体验。

与之前 TDA4 家族不同的是，TDA4VEN 和 AM62P 集成了 K3-DSS (Ultra Lite) 硬件，显示架构有所不同。TDA4VEN/AM62P 采用相同的 K3-DSS (Ultra Lite) 架构，两者都支持三路独立显示输出，通过 OLDI (LVDS)、MIPI DSI 或 DPI (RGB 并行接口) 等多种接口组合，可灵活适配不同应用场景的多屏需求。TDA4VEN 更侧重于汽车应用的功能安全 (支持 ASIL-B)，而 AM62P 则同时面向工业和汽车领域。

本文将深入介绍 K3 DSSUL 显示架构在 TDA4VEN 和 AM62P 的显示子系统架构，阐述如何在 TDA4VEN 和 AM62P 平台上实现同时驱动三个独立显示屏的技术方案，由于默认的 SDK 并没有提供这个例程，同时 Linux 一个显示硬件也无法直接控制两个独立的显示，本文会基于评估版详述基于 Linux 环境下 Weston 和 Kmscube 的例程，给需要 3 屏独立显示方案的用户提供参考和指导。

修改记录

Version	Date	Author	Notes
1.0	April 2025	Fredy Zhang, Jenney Ni	First release

目录

1.	系统框架.....	3
1.1.	嵌入式系统组成	3
1.2.	窗口管理（Shell）	4
1.3.	K3 -DSS UL 介绍	4
2.	如何实现三屏显示的方案.....	5
3.	Demo 的实现例程	6
3.1.	TDA4VEN 硬件连接与配置	6
3.2.	软件准备	7
3.3.	基于 Linux 的 Weston 例程.....	8
3.4.	基于 Linux 的 Kmscube 例程	9
4.	总结.....	11
5.	参考.....	11

图

图 1.	Linux Weston 系统框架.....	3
图 2.	K3 – DSS UL 框图.....	4
图 3.	K3 – DSS UL 框图.....	5
图 4.	SK – LCD1 底部	7
图 5.	TDA4VEN 接口	7
图 6.	Weston 例程效果展示	9
图 7.	Kmscube 例程效果展示	10

1. 系统框架

1.1. 嵌入式系统组成

一个嵌入式 Linux 系统里面常常包含 GPU，DSS 等硬件。典型的 Linux Weston 系统框架如图 1 所示：

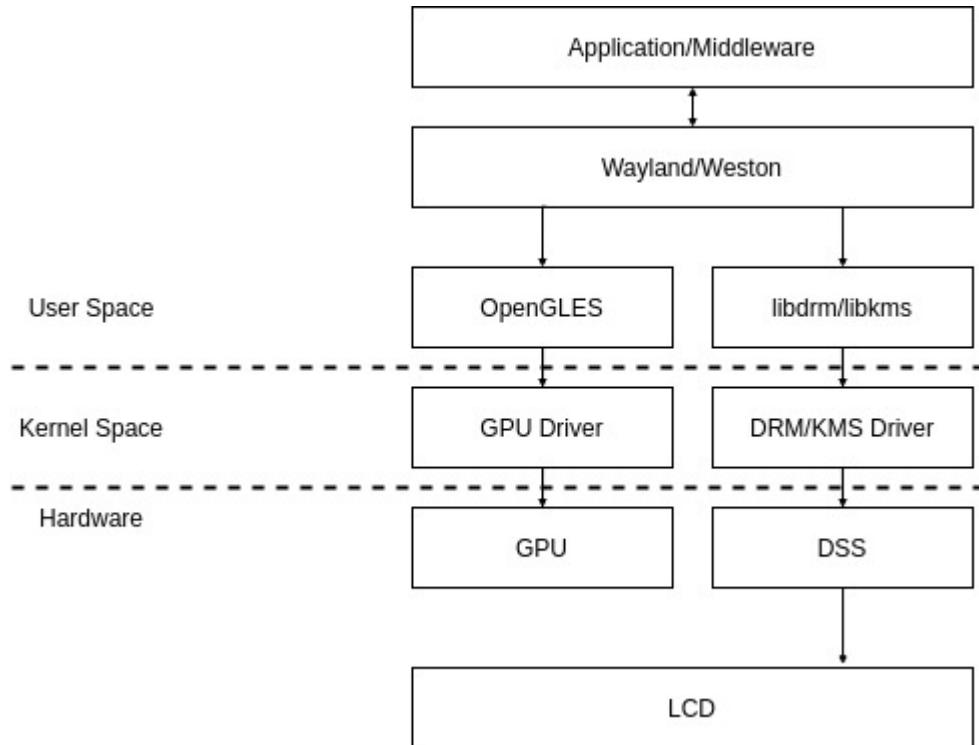


图 1. Linux Weston 系统框架

从应用到硬件，从 User Space 到 Hardware，包含如下模块：

- 应用程序/中间件（用户空间）：Wayland 客户端应用程序或中间件依赖于 Wayland 协议。该客户端可以是传统应用程序或客户端。
- Wayland/Weston（用户空间）：实现 Wayland 显示服务器协议的 Wayland 库，以及用于在 Linux 内核 KMS 上运行显示服务器的 Weston 库。该组件基于 OpenGL ES 和 DRM，可对 Linux 输入设备进行管理。
- OpenGL ES（用户空间）：Weston 用于基于 GPU 合成的 OpenGL ES 库。
- Libdrm/libkms（用户空间）：Weston 合成器使用 libdrm 和 libkms 库来配置显示路径和刷新显示内容。
- GPU 驱动程序（内核空间）：Linux 内核驱动程序，用于将 OpenGL ES 命令传输到 GPU 硬件模块。
- DRM/KMS 驱动程序（内核空间）：DRM/KMS Linux 内核驱动程序，用于访问显示硬件 DSS。
- GPU（硬件）：GPU 硬件模块。
- DSS（硬件）：Display Sub System，显示硬件子系统。
- LCD（硬件）：连接 DSS 的显示器。

1.2. 窗口管理 (Shell)

当前我们有三个窗口管理 shell: Desktop-shell, Fullscreen-shell 和 IVI-shell, 每个 shell 都有自己的公共的客户端协议接口。客户端程序必须按照一种 shell 协议编写客户端, 否则它将无法正常工作。

- **Desktop Shell:** 像现代 X 桌面环境一样，专注于传统的键盘和鼠标用户界面以及熟悉的类似桌面的窗口管理。Desktop-shell 由 desktop-shell.so 和 weston-desktop-shell 插件组成。
 - **Fullscreen Shell:** fullscreen-shell 适用于客户端想要接管所有的输出的场景。客户端在 weston 运行一个 compositor。另外一个 compositor 不需要处理任何特定于平台的内容，例如 DRM/KMS 或 evdev/libinput。fullscreen-shell 仅由 fullscreen-shell.so 插件组成。
 - **IVI shell :** IVI (in-vehicle infotainment)，IVI-shell 是一种专用的 shell，它向控制模块提供了 GENIVI 层管理兼容的 API，对客户端来讲，这是一种简单的协议。IVI-shell 首先加载 ivi-shell.so 插件，然后加载 ivi-controller.so 插件。

针对上述三种 shell，Desktop-shell 最常用，适用于传统的键盘和鼠标的用户界面管理，TI 默认的 SDK 采用 Desktop Shell。Fullscreen Shell 比较少用，适用于客户端想要接管所有的输出的场景，它的一个实际用例是屏幕录制和共享。IVI-Shell 为客户端提供了简单的控制接口，使用者可以方便地对图层及其显示内容的上下层关系、透明度及位置关系等进行控制，为多窗口的应用开发带来了便捷性。因此，在汽车 IVI 和 ADAS 应用中，客户普遍采用了 IVI-shell。

1.3. K3-DSS UL 介绍

K3 显示子系统 (DSS) 是 TI 为其 K3 架构处理器（包括 TDA4x 和 AM62x 系列）设计的统一、可扩展的显示控制器。其核心设计目标是提供灵活的显示输出能力，支持多种接口和格式，并优化功耗。**显示子系统 (DSS)** 是一个硬件模块，TDA4VEN/AM62P 都采用 TI 的 Keystone 3 架构，负责从内存中获取像素数据并将其发送到显示外设，例如 LCD 面板或 DisplayPort 显示器。DSS 硬件可分为两大部分：

- 显示控制器 Display Controller (DISPC)，负责获取像素数据、进行颜色转换、合成和其他像素操作；
 - 外设 Peripherals，负责将原始像素数据编码为标准显示信号，例如 MIPI DPI 或 DSI。支持的显示接口包括：
 - DPI (并行 RGB 接口)：用于驱动传统的 LCD 屏幕。
 - DSI (显示串行接口)：用于连接移动行业处理器接口 (MIPI) DSI 面板，常见于更现代、更紧凑的屏幕。
 - OLDI (LVDS)：用于驱动 LVDS 屏幕，常见于工业面板和汽车仪表盘。

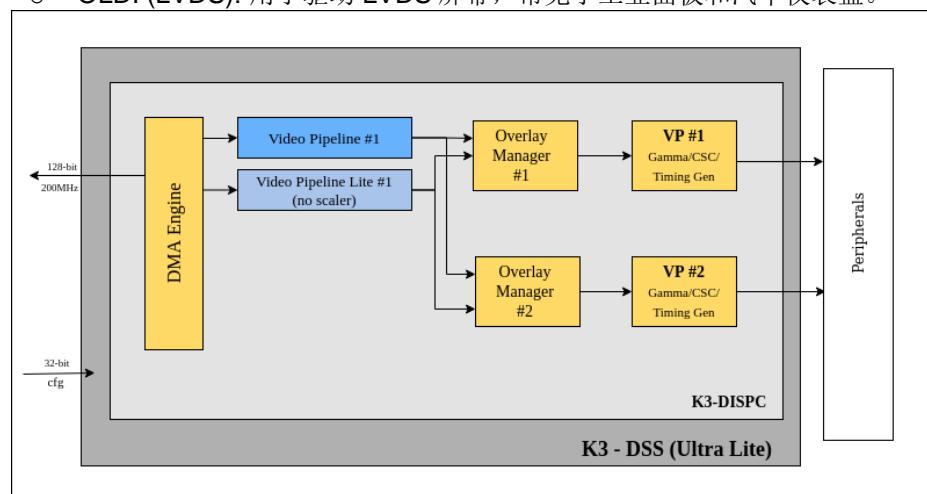


图 2. K3 – DSS UL 框图

除了 SoC 的 DSS 之外，开发板通常还包含外部显示桥接器（例如 DPI 转 HDMI 桥接器）和显示面板。

TDA4VEN/AM62P SoC 内部有两个 DSS7-UL 硬件，分别连接到不同的显示外设。其显示能力

Display Interface	Number of Pins Max	Bandwidth	Applications	Clock
DPI	RGB 24 8 bits/color 4x clock	24 bits x 165MHz = 3.96GB/Sec	Low resolution display systems	DPI (24 位 RGB 并行接口) : 60fps 时高达 1920 x 1080 (165MHz 像素时钟)
LVDS	Single link: 4x Data/1x Clock Dual link: 8x Data/2x Clock	6 x 165MHz x 2 = 1.98 GB/Sec; Up to 3.125 GB/Sec	High resolution displays and industrial applications	OLDI-SL (单链路) : 60fps 时高达 1920 x 1080 (165MHz 像素时钟) OLDI-DL (双链路) : 60fps 时高达 3840 x 1080 (150MHz 像素时钟)
DSI	data pairs and 1 clock pair	4 x 1.5 GB/Sec = 6 GB/Sec	Appliances and medical	MIPI® DSI: 具有 4 通道 MIPI® D-PHY, 在 60fps 下支持高达 3840 x 1080 (300MHz 像素时钟)

2. 如何实现三屏显示的方案

实现三屏显示的核心在于合理分配两个 DSS 硬件的 DSPC。由于每个显示硬件有两个独立的显示控制器，要实现三个物理屏同时显示不同内容的效果，需要采用一种“一拖二”的策略。每个显示控制器 DSPC 有两个 Video pipeline 也给了“一拖二”的可能性。

如下图所示，一个 video pipeline 1 链接一个 overlay Manager1，然后连接 Video Port1，最后连接一个硬件显示接口。同理，video pipeline 2 链接一个 overlay Manager2，然后连接 Video Port2，最后连接另外一个硬件显示接口。由于内容来自不同的 Video Pipeline，因此，这两个屏幕就可以显示不同的内容。

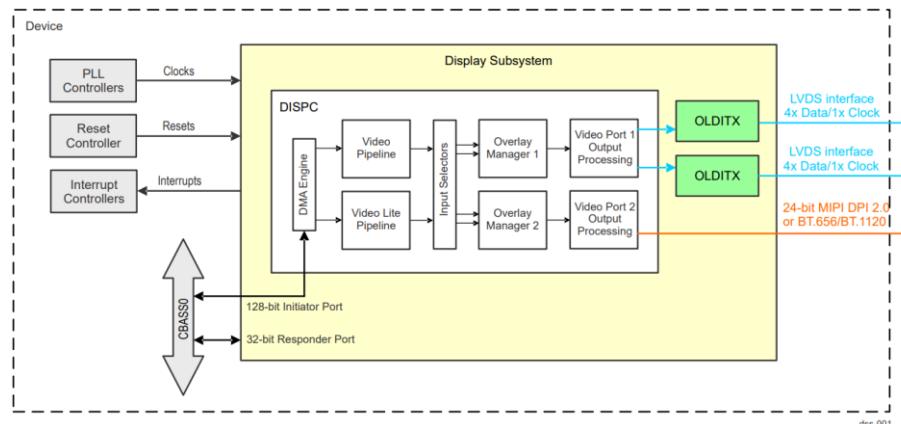


图 3. K3 – DSS UL 框图

如下图所示，TDA4VEN 使用了两个 K3-DSS UL 的架构，系统内部显示连接如下。AM62P 采用类似的架构，参考 TRM 的内容。

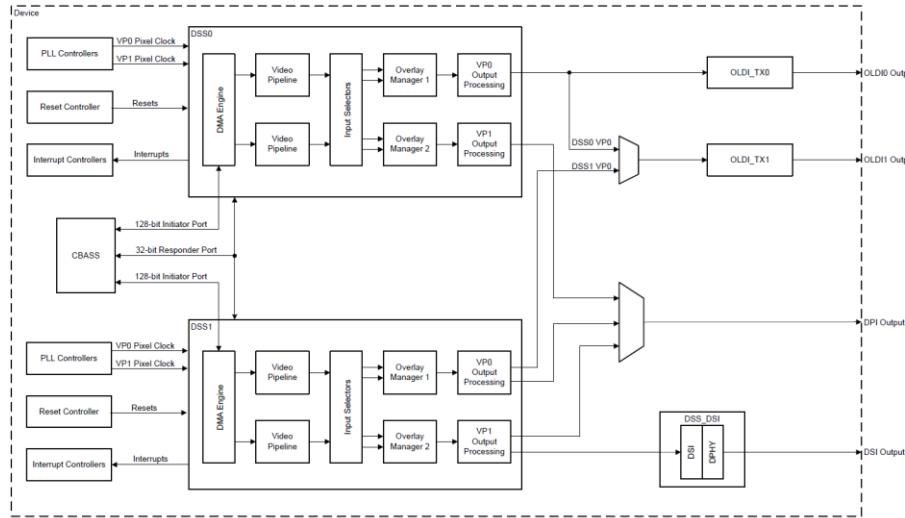


图 4. TDA4VEN 显示框图

依据硬件的情况，所以利用 DSS0 和 DSS1 我们的三屏显示方案是：

- **DSS0**： 屏幕 1 和屏幕 2 连接到同一个 DSPC，每个 Video pipeline 独立控制一个 Video Port。
- **DSS1**： 屏幕 3 独立控制一个显示硬件，控制一个显示 Port。

由于默认的 Linux 系统中，一个显示硬件控制两个独立显示，本文的第三部分讲会详细介绍如何实现上述的方案。

3. Demo 的实现例程

以下是基于 Linux 的 Weston 和 Kmscube 例程，Weston 是基于主流的图形化框架，适合使用 Weston 的框架的用户，对于没有采用的 Weston 框架的用户，Kmscube 有源码提供，用户可以根据这部分的源码进行修改。本文基于 TI 的默认的 Processor SDK 来进行，方便用户直接基于评估板来测试。具体硬件配置和软件的示例如下：

3.1. TDA4VEN 硬件连接与配置

依据 TDA4VEN 的实际情况，我们硬件配置是：

- 屏幕 1： 21.5 英寸 LG 台式显示屏，显示分辨率 1920x1080，接口是 HDMI；
- 屏幕 2： 10.1 英寸 SK-LCD1，显示分辨率 1920x1200，接口是 OLDI(LVDS)；



图 5. SK – LCD1 底部

- 屏幕 3: Raspberry Pi 7 Inch Touch Screen Display, 显示分辨率 800x480, 接口是 DSI; 系统连接接口如下:

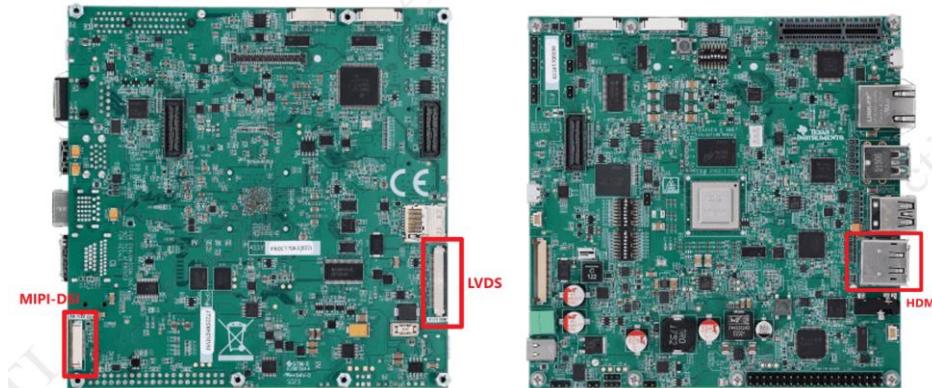


图 6. TDA4VEN 接口

Note: Raspberry Pi 屏与 TDA4VEN 的 DSI 连接需要 22PIN 转 15PIN 的排线, 或者两者之间另接转接板。

3.2. 软件准备

TDA4VEN 软件和使用指导链接: [软件](#) 和 [使用指导](#);

1. 参考 TDA4VEN 的开发使用指导, 并按照指导安装好环境, 首先在板子上安装默认 SD 卡启动镜像 (这里采用 11_01_00_03 SDK, TI 采用统一标准的 Linux SDK, AM62x 采用相似的方法, 最新的 SDK 仍旧适用);

`PC_ENV_Linux_SDK # sudo ./create_sd.sh`

https://software-dl.ti.com/jacinto7/esd/processor-sdk-linux-j722s/11_01_00_03/exports/docs/devices/J7_Family/linux/index.html

2. (一次操作) 在 SD 卡的 boot 分区中, 找到 uEnv.txt 文件; 注释掉 VISION_APPS 的 overlay, 并且增加如下 overlays :

```
# dropprocboot=1  
# name_overlays=ti/k3-j722s-vision_apps.dtbo  
name_overlays=ti/k3-j722s-evm-dsi-rpi-7inch-panel.dtbo ti/k3-j722s-evm-microtips-mf101hie-panel.dtbo
```

在 MCU2_0 中禁用 VISION_APPS 的显示或删除指向 VISION_APPS 二进制的软链接。

3. （每次启动都需要执行或者在环境里设置好变量并保存）也可以在 U-boot 启动期间，禁用 VISION_APPS；在 U-BOOT 启动过程中，输入任意键，进入 U-BOOT 命令行，输入如下命令：

```
U-boot # setenv name_overlays ti/k3-j722s-evm-dsi-rpi-7inch-panel.dtbo ti/k3-j722s-evm-microtips-mf101hie-panel.dtbo  
U-boot # boot
```

4. 然后正常启动设备到文件系统，并能看到命令行界面。

3.3. 基于 Linux 的 Weston 例程

Note: Weston 例程需要连接鼠标，具体接口参考 TDA4VEN 开发使用指导。

1. 在 shell 窗口输入 Weston 命令，点亮所有屏幕：

```
root@j722s-evm:~# weston --backends drm --drm-device=card1 --additional-devices=card2
```

2. 分别使用如下三个命令，在三个屏幕上启动三个不同的 Wayland 显示应用：

```
a. gst-launch-1.0 videotestsrc ! waylandsink --no-position  
b. glmark2-es2-wayland  
c. weston-terminal
```

Note: 默认在鼠标所在的屏幕上启动 Wayland 显示应用。需要连接鼠标并移动到不同的屏幕上，然后启动终端，输入命令后在不同屏幕上启动应用程序。

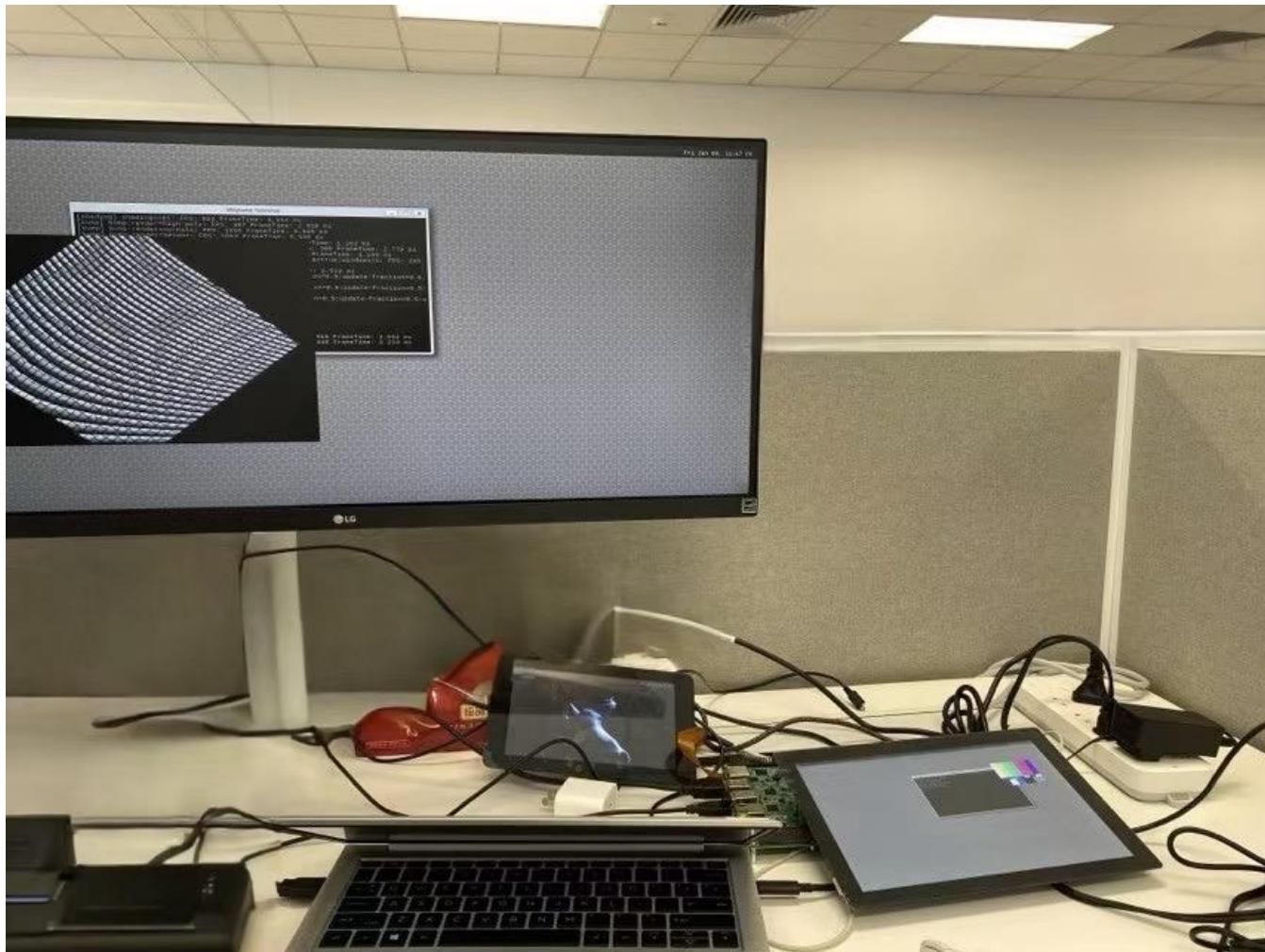


图 7. Weston 例程效果展示

3. 也可以在 Wayland 应用程序中选择不同的 wl_outputs，在不同的屏幕上打开相应显示应用，以下命令可用于列出所有可能的输出屏幕：

```
root@j722s-evm:~# wayland-info -i wl_output
```

3.4. 基于 Linux 的 Kmscube 例程

1. 首先，修改 Linux_SDK 中的 drm_ioctl.h，释放对 DRM 的权限，这样就可以一个显示硬件控制两个屏幕。

```

From 0882eef8a3fc680a62aa8d3875845da8b1a7831f Mon Sep 17 00:00:00 2001
From: Rahul T R <r-ravikumar@ti.com>
Date: Tue, 11 Feb 2025 00:54:55 +0530
Subject: [PATCH] HACK: drm/drm_ioctl: Remove requirement for permission for all
        drm IOCTLs

Signed-off-by: Rahul T R <r-ravikumar@ti.com>
---
include/drm/drm_ioctl.h | 2 ++
1 file changed, 1 insertion(+), 1 deletion(-)

diff --git a/include/drm/drm_ioctl.h b/include/drm/drm_ioctl.h
index 6ed61c371..41931c74d 100644
--- a/include/drm/drm_ioctl.h
+++ b/include/drm/drm_ioctl.h
@@ -97,7 +97,7 @@ enum drm_ioctl_flags {
     * unauthenticated clients, or when a master is not the currently active
     * one.
 */
- DRM_MASTER      = BIT(1),
+ DRM_MASTER      = 0,
 */
 * @DRM_ROOT_ONLY:
 *
-- 2.34.1

```

2. 重新编译 Linux 镜像，并且安装到 SD 卡上:

PC_ENV_Linux_SDK # make linux

PC_ENV_Linux_SDK # make linux_install ROOT_PART=/media/jenney/root

3. 使用如下命令，在不同屏幕上运行 Kmscube:

```

kmscube -D /dev/dri/card0 -n 0 // 点亮 Display 1
kmscube -D /dev/dri/card0 -n 1 // 点亮 Display 2
kmscube -D /dev/dri/card1 -n 0 // 点亮 Display 3

```

kmscube -D /dev/dri/card0 -n 0 & kmscube -D /dev/dri/card0 -n 1 & kmscube -D /dev/dri/card1 -n 0 // 同时点亮三个 Displays

Note: 这是一个 HACK 方法，也可以在显示应用程序中使用 drmDrop/SetMaster drmDropMaster(fd) APIs，实现相同效果。



图 8. Kmscube 例程效果展示

4. 总结

本文通过软硬件结合分享了三屏显示的系统，即可在 TI TDA4VEN 和 AM62P 平台上实现灵活的三屏显示系统。在实际应用中，我们发现越来越多的应用需要多个屏幕的独立显示，希望这篇文章可以帮助到 TDA4VEN 和 AM62P 的用户实现多屏显示方案。

5. 参考

1. <https://manpages.debian.org/experimental/weston/weston.1.en.html>
2. [Top Five Design Considerations for Smart](#)
3. [Display Interfaces: A Comprehensive Guide to Sitara MPU](#)
4. <https://wayland.freedesktop.org/>
5. <https://at.projects.genivi.org/wiki/display/PROJ/Wayland+IVI+Extension+Design>
6. <http://software-dl.ti.com/processor-sdk-linux/esd/docs/latest/linux/index.html>
7. https://www.ti.com/cn/lit/ds/symlink/tda4vm.pdf?ts=1618976597409&ref_url=https%253A%252F%252Fwww.ti.com.cn%252Fproduct%252Fcn%252FTDA4VM
8. <https://lwn.net/Articles/618197/>

重要通知和免责声明

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、与某特定用途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安全、安保法规或其他要求。

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#))、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 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。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2025 , 德州仪器 (TI) 公司

最后更新日期 : 2025 年 10 月