

Application Note

在 AM62P 上集成 LVDS 面板



Suren Porwar, Soumya Tripathy

Sitara MPU

摘要

本应用手册介绍了如何将定制低电压差分信号 (LVDS) 或开放 LVDS 显示接口 (OLDI) 面板与 AM62P Sitara 片上系统 (SoC) 上基于 Linux® 的系统和实时操作系统 (RTOS) 相集成。LVDS 面板通常用于显示器中，将定制面板与 Linux 集成需要配置内核、器件树，还可能需编写显示驱动程序。本文档概述了在 AM62P 上的定制 LVDS 面板与 Linux 系统或 RTOS 之间实现正确通信的必要步骤。

内容

1 引言.....	1
2 AM62Px 处理器.....	2
2.1 主要特性和优势.....	2
2.2 AM62P 显示子系统.....	3
2.3 AM62P 显示子系统时钟架构.....	3
3 显示信号和时序参数.....	5
4 集成步骤 (Linux®).....	5
4.1 Linux® 叠加层文件.....	5
4.2 时序信息.....	7
5 集成步骤 (RTOS).....	8
6 术语.....	9

插图清单

图 2-1. AM62P 方框图.....	2
图 2-2. DSS 方框图.....	3
图 2-3. PLL18 在双链路模式或单链路模式下为 OLDI TX 提供时钟输入.....	4
图 2-4. 使用 K3 配置工具的 DSS0 时钟.....	5
图 5-1. SysConfig 配置器.....	8

商标

Android™ is a trademark of Google Inc.

Linux® is a registered trademark of Linus Torvalds.

Arm® and Cortex® are registered trademarks of Arm Limited.

MIPI® is a registered trademark of Mobil Industry Processor Interface Alliance.

所有商标均为其各自所有者的财产。

1 引言

由于具有高速数据传输功能和抗噪能力，LVDS 面板广泛用于各种电子设备，包括嵌入式系统、笔记本电脑和工业显示器。将定制 LVDS 面板与基于 Linux 的系统进行集成涉及配置内核以支持该面板的分辨率、时序和通信协议。

2 AM62Px 处理器

AM62Px (P 表示加强版) 是现有 Sitara™ AM62x 低成本系列应用处理器的扩展, 专为高性能嵌入式 3D 显示应用而构建。可扩展的 Arm® Cortex®-A53 性能和嵌入式功能 (例如多屏幕高清显示支持、3D 图形加速、4K 视频加速和广泛的外设) 使 AM62Px 非常适合广泛的汽车和工业应用, 包括汽车数字仪表、汽车显示器、工业 HMI 等。

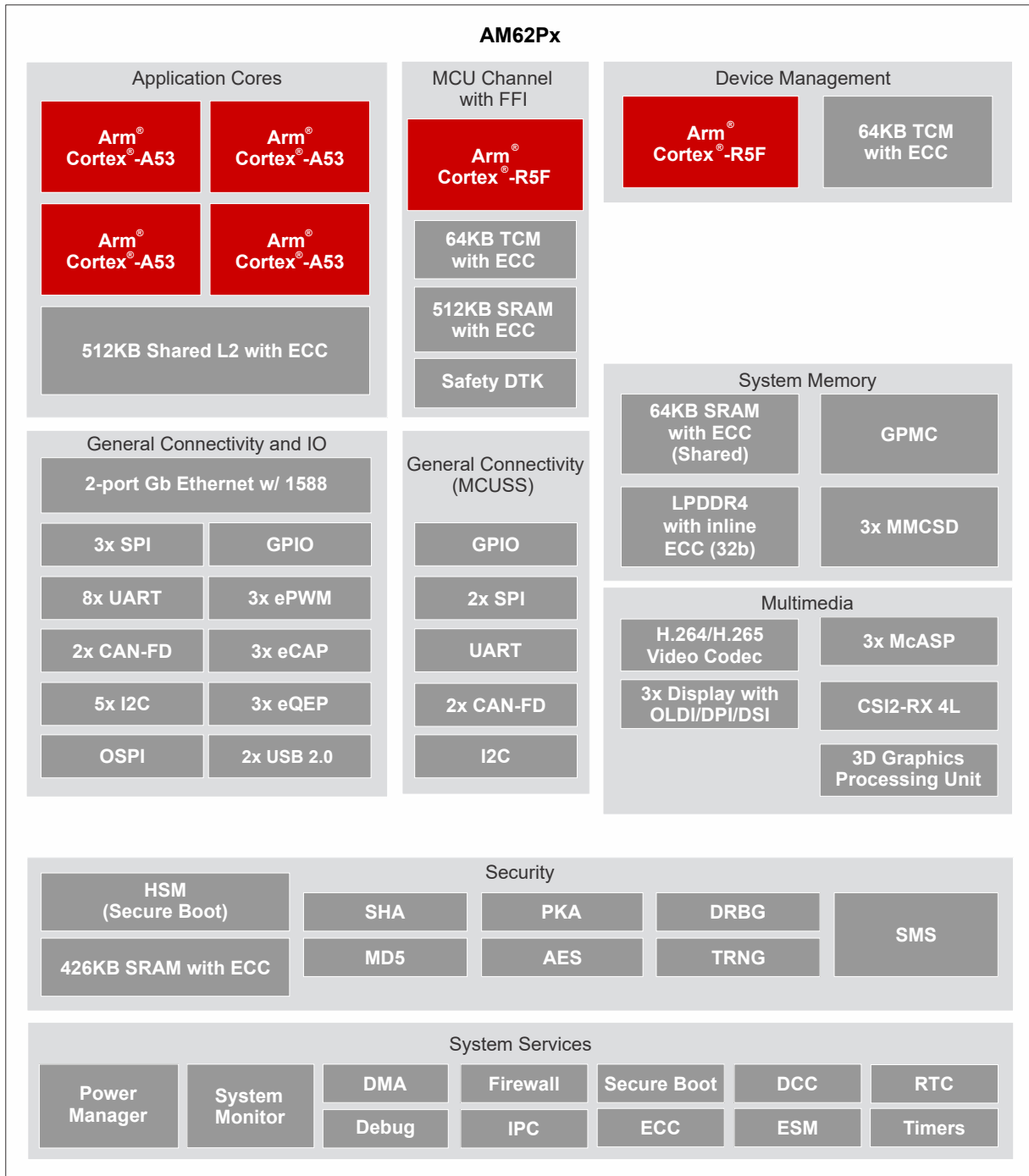


图 2-1. AM62P 方框图

2.1 主要特性和优势

AM62Px 处理器具有以下特性和优势：

- 借助 Linux® 和 Android™ SDK 以及实时功能安全和信息安全 SDK 专注于创新和快速开发。
- 利用通过 OLDI (LVDS) (1 个 OLDI-DL、1 个或 2 个 OLDI-SL)、DSI 或 DPI 实现的三路显示以及新一代 3D 图形处理单元 (GPU) 和 4K 视频加速，满足下一波 HMI 设计的需求。
- 使用 Arm Cortex-A53 中央处理单元 (CPU) 及开源 AI 软件和工具提供智能功能，例如面部识别和非接触式 HMI。

2.2 AM62P 显示子系统

AM62P DSS 具有以下特性：

- 通过 OLDI (LVDS) (1 个 OLDI-DL、1 个或 2 个 OLDI-SL)、DSI 或 DPI 实现三路显示支持
 - OLDI-SL：60fps 时高达 1920 x 1080 (165MHz 像素时钟)
 - OLDI-DL：60fps 时高达 3840 x 1080 (150MHz 像素时钟)
 - 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)
 - DPI (24 位 RGB 并行接口)：60fps 时高达 1920 x 1080 (165MHz 像素时钟)
- 具有硬件叠加支持的四条显示流水线。每个显示屏最多可使用两条显示流水线。
- 支持定帧检测和数据正确性检查等安全功能。

图 2-2 展示了 DSS 方框图。

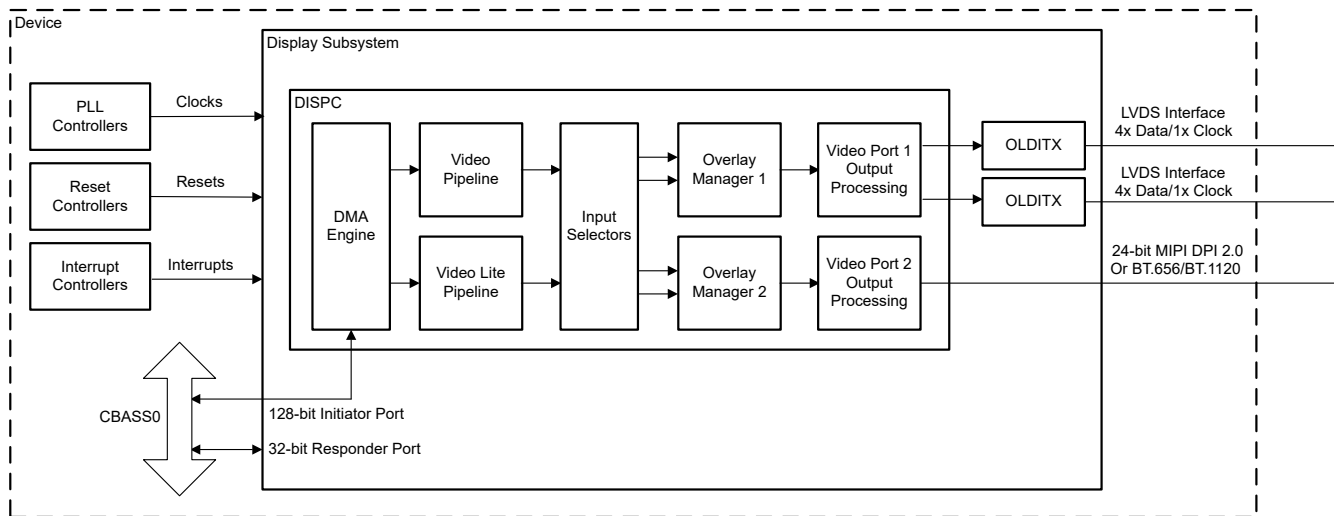


图 2-2. DSS 方框图

2.3 AM62P 显示子系统时钟架构

AM62Px 系列 SoC 有两个 DSS 控制器实例 (DSS0 和 DSS1)。这两个 DSS 控制器由三个 PLL 支持，因此最多可以有三个独立的视频数据流，即 OLDI、DPI 和 DSI。

然后，可以在克隆模式下配置两个 OLDI，使其具有一对 (两个克隆显示屏 + 另外两个独立的显示屏)，从而为四个视频接收器提供总计输出。有两个由 VP0 处理 (DSS0) 的 OLDI TX 以及一个来自 VP1 的 DPI 输出。DSS1 VP0 只能输出到两个可用的 OLDI TX 中的一个，从而允许单链路模式。但是，DSS1 VP0 也可以绕过 OLDI TX，只是在 SoC 上提供 DPI 输出。

DSS1 VP1 还可提供 DPI 输出。除此之外，DSS1 VP1 还可将输出重定向到 DSI 控制器，并帮助从 SoC 发出 DSI 信号。

有三个 VP (DSS0-VP1、DSS1-VP0、DSS1-VP1) 能够驱动从 SoC 发出的 DPI 信号，但在给定的时间只有一个 VP 能够驱动这些信号。所有这些 VP 的 DPI 输出都通过多路复用器传递，SoC 只发出一组 DPI 信号。

PLL16 是专用的 OLDI PLL，仅与 DSS0-VP0 配合使用。此 PLL 有助于为双链路模式运行提供最高 2.1GHz 的频率。

PLL17 是 DPI PLL，为所有三个支持 DPI 输出的 VP (DSS0-VP1、DSS1-VP0 和 DSS1-VP1) 提供像素时钟。最大频率为 165MHz。在一种情况下，PLL17 为 DSS1-VP1 提供 300MHz 像素频率以驱动 DSI 输出。当使用 DSI PLL (PLL18) 从 DSS1-VP0 驱动单链路输出时，会发生这种情况。

PLL18 兼作 DSI 或 OLDI 时钟。出于 DSI 目的，PLL18 为 DSS1-VP1 提供一个最大频率为 300MHz 的时钟。出于 OLDI 目的，PLL18 支持用于单链路 OLDI 输出的 DSS1-VP0。

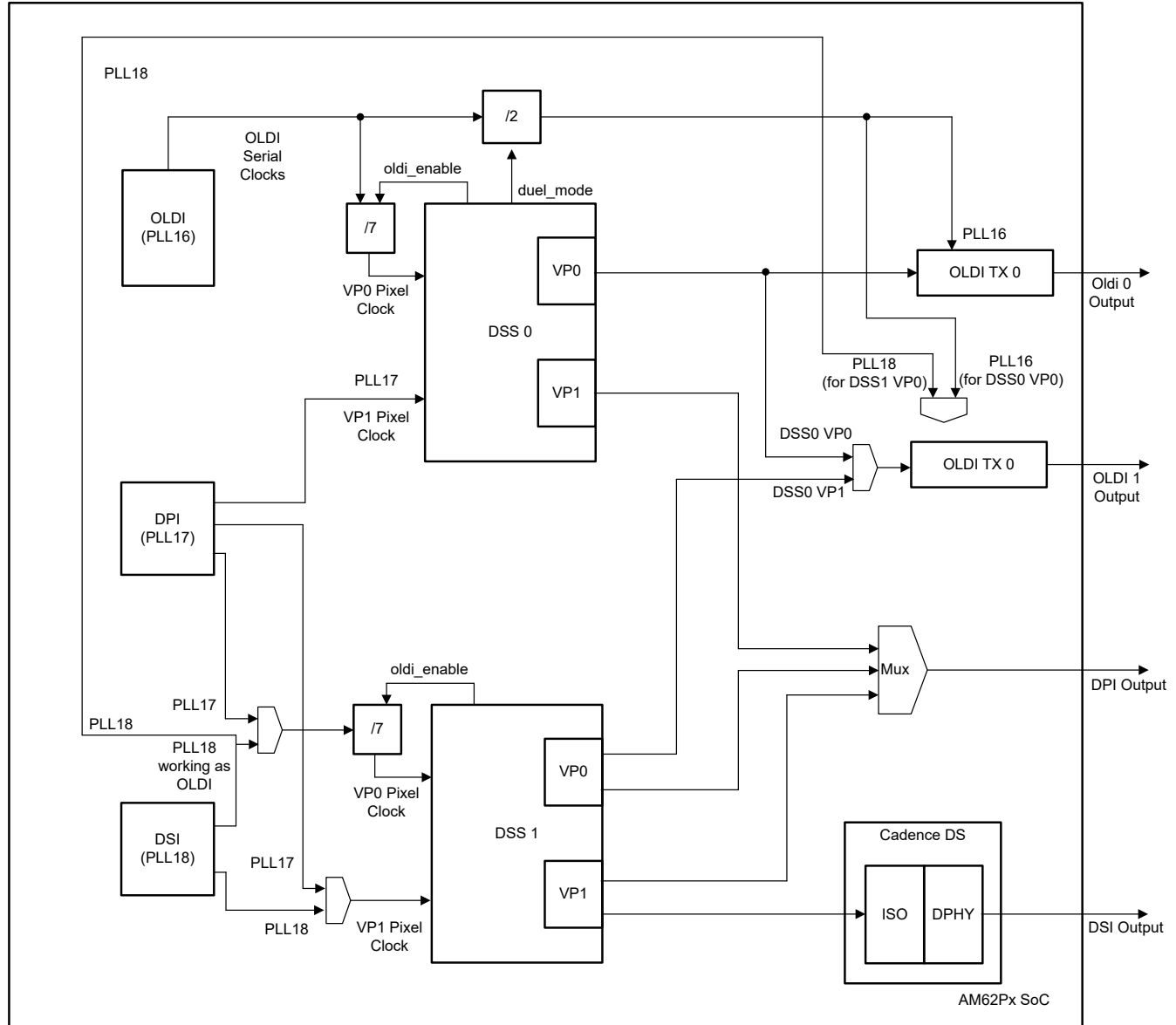


图 2-3. PLL18 在双链路模式或单链路模式下为 OLDI TX 提供时钟输入

PLL16 在双链路模式或单链路模式下为 OLDI TX 提供时钟输入。在 DSS0 VP0 中启用 `oldi_enable_cfg0` 会将输入 PLL16 时钟进行七倍分频作为 DSS0 VP0 像素时钟的输入。在 DSS0 VP0 中启用 `dual_cfg0` 会将输入 PLL16 时钟进行分频，分频倍数是 OLDI TX 的 PLL 时钟的二倍。

PLL18 在单链路模式下为 OLDI TX1 提供时钟输入。在 DSS1 VP0 中启用 `oldi_enable_cfg0` 会将输入 PLL8 时钟进行 7 倍分频作为 DSS1 VP0 像素时钟的输入。

图 2-4 展示了使用 k3conf 工具的 DSS0 时钟屏幕截图，该工具也称为 K3conf 时钟转储片段。

```
root@am62pxx-evm:~# k3conf dump clocks | grep 'DSS0\|OLDI'
186      0      | DEV_DSS0_DPI_0_IN_CLK | CLK_STATE_READY | 1050000000
186      2      | DEV_DSS0_DPI_1_IN_CLK | CLK_STATE_READY | 3000000000
186      3      | DEV_DSS0_DPI_1_IN_CLK_PARENT_HSDIV0_16FFT_MAIN_17_HSDIVOUT0_CLK | CLK_STATE_READY | 3000000000
186      4      | DEV_DSS0_DPI_1_IN_CLK_PARENT_BOARD_0_VOUT0_EXTFCLKIN_OUT | CLK_STATE_READY | 0
186      5      | DEV_DSS0_DPI_1_OUT_CLK | CLK_STATE_READY | 0
186      6      | DEV_DSS0_DSS_FUNC_CLK | CLK_STATE_READY | 3200000000
234      0      | DEV_OLDI_TX_CORE0_OLDI_0_FWD_P_CLK | CLK_STATE_READY | 1050000000
234      5      | DEV_OLDI_TX_CORE0_OLDI_PLL_CLK | CLK_STATE_READY | 1050000000
235      0      | DEV_OLDI_TX_CORE1_OLDI_0_FWD_P_CLK | CLK_STATE_READY | 1050000000
235      1      | DEV_OLDI_TX_CORE1_OLDI_0_FWD_P_CLK_PARENT_HSDIV0_16FFT_MAIN_16_HSDIVOUT0_CLK | CLK_STATE_READY | 1050000000
235      2      | DEV_OLDI_TX_CORE1_OLDI_0_FWD_P_CLK_PARENT_MAIN_DSS1_DPIO_PCLK_OUT0 | CLK_STATE_READY | 3000000000
235      7      | DEV_OLDI_TX_CORE1_OLDI_PLL_CLK | CLK_STATE_READY | 1050000000
235      8      | DEV_OLDI_TX_CORE1_OLDI_PLL_CLK_PARENT_HSDIV0_16FFT_MAIN_16_HSDIVOUT0_CLK | CLK_STATE_READY | 1050000000
235      9      | DEV_OLDI_TX_CORE1_OLDI_PLL_CLK_PARENT_HSDIV0_16FFT_MAIN_18_HSDIVOUT0_CLK | CLK_STATE_READY | 3000000000
root@am62pxx-evm:~#
```

图 2-4. 使用 K3 配置工具的 DSS0 时钟

应用手册的以下各部分提供了在 Linux 和 RTOS 上集成 LVDS 面板的要求和步骤。

3 显示信号和时序参数

[DIGI 嵌入式文档门户](#) 网页提供了 *LCD 显示信号和时序参数* 的定义和说明。这些定义对 OLDI 也有效。

每个制造商提供显示时序的方式略有不同，有些制造商提供的时序比其他制造商提供的时序更详细。

收集所有分辨率和时序参数后，使用 AM62P 将它们集成到 Linux OS 或 RTOS 中。

4 集成步骤 (Linux®)

要从 AM62P 的角度支持显示，请完成以下操作：

1. 通过定义显示叠加层来更新器件树，或使用标出特定面板的显示条目直接修改器件树。
2. 将显示时序添加到具有相同兼容性字符串的 `panel-simple.c`。

Linux 驱动程序中启用了 SK-LCD1 支持。节 4.1 和节 4.2 中的代码片段分别演示了如何使用 `panel-simple.c` 创建 Linux 叠加层文件和时序信息。

有关更多信息，请参阅以下链接：

- <https://www.kernel.org/doc/Documentation/devicetree/bindings/display/panel/panel-simple.yaml>

4.1 Linux® 叠加层文件

本节中的代码片段由以下参考资料提供支持：

- <https://git.ti.com/cgi/ti-linux-kernel/ti-linux-kernel/tree/arch/arm64/boot/dts/ti/k3-am625-sk-microtips-mf101hie-panel.dtso?h=ti-linux-6.1.y>

```
// SPDX-License-Identifier: GPL-2.0
/**
 * Microtips integrated OLDI panel (MF-101HIEBCAF0) and touch DT overlay for AM625 - SK
 * Copyright (C) 2021 Texas Instruments Incorporated - http://www.ti.com/
 */

/dts-v1/;
/plugin/;

#include <dt-bindings/gpio/gpio.h>
#include <dt-bindings/interrupt-controller/irq.h>

#include "k3-pinctrl.h"

&{/} {
    display {
        compatible = "microtips,mf-101hiebcaf0";
        /*
         * Note that the OLDI TX 0 transmits the odd set of pixels
         * while the OLDI TX 1 transmits the even set. This is a
         * fixed configuration in the IP integration and is not
         * changeable. The properties, "dual-lvds-odd-pixels" and
         * "dual-lvds-even-pixels" have been used to merely
         * identify if a Dual Link configuration is required.
         * Swapping them will not make any difference.
         */
    }
}
```

```

*/
port@0 {
    dual-lvds-odd-pixels;
    lcd_in0: endpoint {
        remote-endpoint = <&oldi_out0>;
    };
};
port@1 {
    dual-lvds-even-pixels;
    lcd_in1: endpoint {
        remote-endpoint = <&oldi_out1>;
    };
};
};
};
};
&main_pmx0 {
    main_olddi0_pins_default: main-olddi0-pins-default {
        pinctrl-single,pins = <
            AM62X_IOPAD(0x0260, PIN_OUTPUT, 0) /* (AA5) OLDDI0_A0N */
            AM62X_IOPAD(0x025c, PIN_OUTPUT, 0) /* (Y6) OLDDI0_A0P */
            AM62X_IOPAD(0x0268, PIN_OUTPUT, 0) /* (AD3) OLDDI0_A1N */
            AM62X_IOPAD(0x0264, PIN_OUTPUT, 0) /* (AB4) OLDDI0_A1P */
            AM62X_IOPAD(0x0270, PIN_OUTPUT, 0) /* (Y8) OLDDI0_A2N */
            AM62X_IOPAD(0x026c, PIN_OUTPUT, 0) /* (AA8) OLDDI0_A2P */
            AM62X_IOPAD(0x0278, PIN_OUTPUT, 0) /* (AB6) OLDDI0_A3N */
            AM62X_IOPAD(0x0274, PIN_OUTPUT, 0) /* (AA7) OLDDI0_A3P */
            AM62X_IOPAD(0x0280, PIN_OUTPUT, 0) /* (AC6) OLDDI0_A4N */
            AM62X_IOPAD(0x027c, PIN_OUTPUT, 0) /* (AC5) OLDDI0_A4P */
            AM62X_IOPAD(0x0288, PIN_OUTPUT, 0) /* (AE5) OLDDI0_A5N */
            AM62X_IOPAD(0x0284, PIN_OUTPUT, 0) /* (AD6) OLDDI0_A5P */
            AM62X_IOPAD(0x0290, PIN_OUTPUT, 0) /* (AE6) OLDDI0_A6N */
            AM62X_IOPAD(0x028c, PIN_OUTPUT, 0) /* (AD7) OLDDI0_A6P */
            AM62X_IOPAD(0x0298, PIN_OUTPUT, 0) /* (AD8) OLDDI0_A7N */
            AM62X_IOPAD(0x0294, PIN_OUTPUT, 0) /* (AE7) OLDDI0_A7P */
            AM62X_IOPAD(0x02a0, PIN_OUTPUT, 0) /* (AD4) OLDDI0_CLK0N */
            AM62X_IOPAD(0x029c, PIN_OUTPUT, 0) /* (AE3) OLDDI0_CLK0P */
            AM62X_IOPAD(0x02a8, PIN_OUTPUT, 0) /* (AE4) OLDDI0_CLK1N */
            AM62X_IOPAD(0x02a4, PIN_OUTPUT, 0) /* (AD5) OLDDI0_CLK1P */
        >;
    };
};
&dss {
    pinctrl-names = "default";
    pinctrl-0 = <&main_olddi0_pins_default &main_dss0_pins_default>;
};
&dss_ports {
    #address-cells = <1>;
    #size-cells = <0>;

    /* VP1: LVDS Output (OLDDI TX 0) */
    port@0 {
        reg = <0>;
        olddi_out0: endpoint {
            remote-endpoint = <&lcd_in0>;
        };
    };

    /* VP1: LVDS Output (OLDDI TX 1) */
    port@2 {
        reg = <2>;
        olddi_out1: endpoint {
            remote-endpoint = <&lcd_in1>;
        };
    };
};
&main_i2c0 {
    #address-cells = <1>;
    #size-cells = <0>;

    touchscreen@41 {
        compatible = "ilitek,ili251x";
        reg = <0x41>;
        interrupt-parent = <&exp1>;
        interrupts = <22 IRQ_TYPE_EDGE_FALLING>;
    };
};

```

```
};
```

4.2 时序信息

本节中的代码片段由以下参考资料提供支持：

- <https://git.ti.com/cgi/ti-linux-kernel/ti-linux-kernel/tree/drivers/gpu/drm/panel/panel-simple.c?h=ti-linux-6.1.y#n2703>

```
static const struct drm_display_mode microtips_mf_101hiebcaf0_mode = {
    .clock = 150275,
    .hdisplay = 1920,
    .hsync_start = 1920 + 32,
    .hsync_end = 1920 + 32 + 52,
    .htotal = 1920 + 32 + 52 + 24,
    .vdisplay = 1200,
    .vsync_start = 1200 + 24,
    .vsync_end = 1200 + 24 + 8,
    .vtotal = 1200 + 24 + 8 + 3,
};

static const struct panel_desc microtips_mf_101hiebcaf0 = {
    .modes = &microtips_mf_101hiebcaf0_mode,
    .bpc = 8,
    .num_modes = 1,
    .size = {
        .width = 217,
        .height = 136,
    },
    .delay = {
        .prepare = 50,
        .disable = 50,
    },
    .bus_flags = DRM_BUS_FLAG_DE_HIGH,
    .bus_format = MEDIA_BUS_FMT_RGB888_1X7X4_SPWG,
    .connector_type = DRM_MODE_CONNECTOR_LVDS,
};
```

5 集成步骤 (RTOS)

使用该 SysConfig 配置器在 RTOS 中集成新的 LVDS 面板，该配置器是 MCU_PLUS_SDK 的一部分。DSS 驱动程序程序的 SysConfig 提供了以下配置选项卡，用于集成新的 LVDS 面板。

1. 视频端口时序配置：
 - 配置面板所需的 **像素时钟频率**
 - 时序参数 HSYNC、VSYNC、VFP、VBP、HFP、HBP 等
 - 接口宽度：24 位或 18 位
 - 时序信号极性
2. OLDI 配置：
 - OLDI 的映射类型：VESA、JEIDA、24 位或 18 位标准
 - 输入位宽：连接的面板是需要 24 位还是 18 位输入
 - 双链路模式：启用或禁用双链路模式
 - 数据使能信号极性

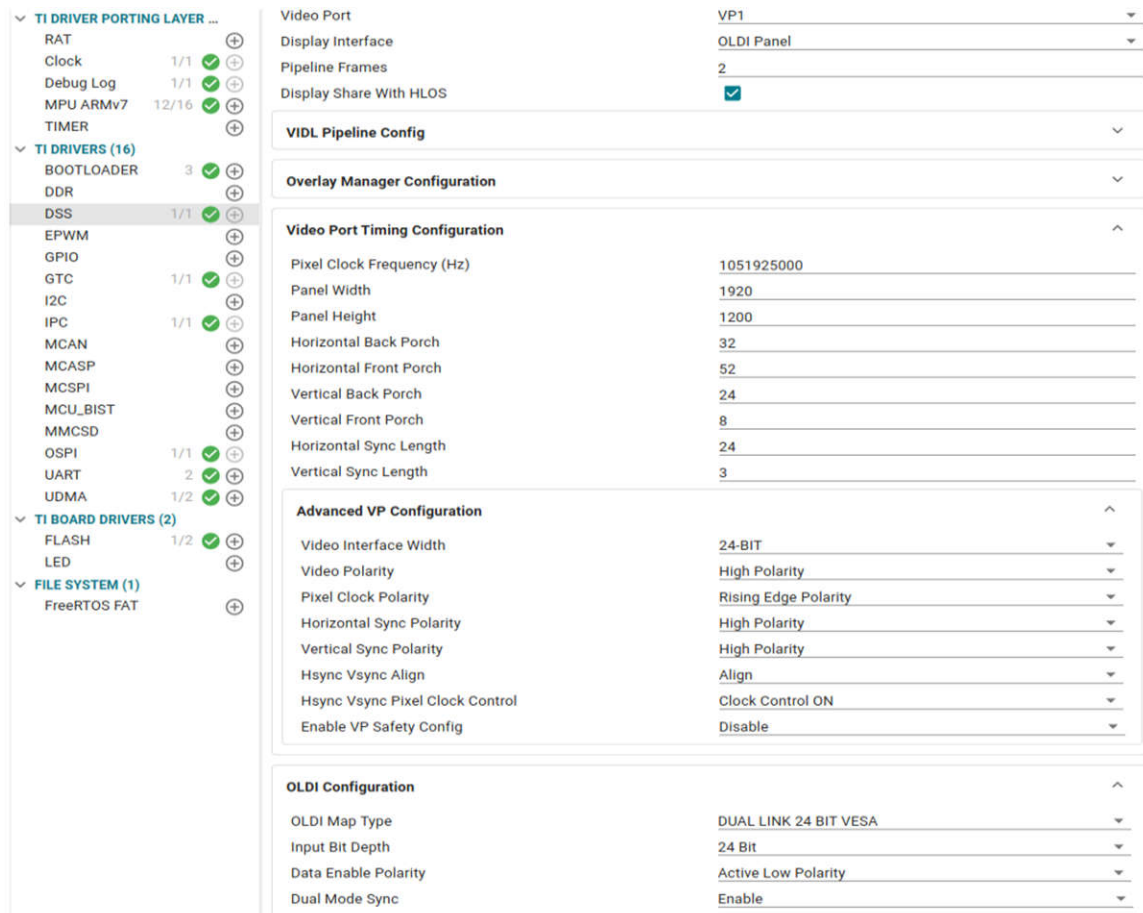


图 5-1. SysConfig 配置器

6 术语

LVDS	低电压差分信号
OLDI	开放式 LVDS 显示接口
RTOS	实时操作系统
SoC	片上系统
HMI	人机界面
SDK	软件开发套件
OLDI-SL	OLDI-单链路
OLDI-DL	OLDI-双链路
DSI	显示串行接口
DPI	显示并行接口
MIPI	移动产业处理器接口
RGB	红色、绿色、蓝色
DSS	显示子系统
PLL	锁相环
HSYNC	水平同步
VSYNC	垂直同步
DRDY	数据就绪、数据使能
DE	数据使能
VBP	垂直后沿
VFP	垂直前沿
HBP	水平后沿
HFP	水平前沿
OS	操作系统

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