Application Note

LVDS Panel Integration on AM62P

Suren Porwar, Soumya Tripathy *Sitara MPU*

ABSTRACT

This application note shows how to integrate a custom low-voltage differential signaling (LVDS) or open LVDS display interface (OLDI) panel with a Linux®-based system and real-time operating system (RTOS) on the AM62P Sitara system-on-a-chip (SoC). LVDS panels are commonly used in displays, and integrating a custom panel with Linux requires configuring the kernel, device tree, and possibly writing a display driver. This document outlines the necessary steps to enable proper communication between the custom LVDS panel on AM62P and the Linux system or RTOS.

Table of Contents

List of Figures

Trademarks

1 Introduction

Due to the high-speed data transmission capabilities and noise immunity, LVDS panels are widely used in various electronic devices, including embedded systems, laptops, and industrial displays. Integrating a custom LVDS panel with a Linux-based system involves configuring the kernel to support the resolution, timing, and communication protocol of the panel.

2 AM62Px Processor

The AM62Px (P = Plus) is an extension of the existing Sitara™ AM62x low-cost family of application processors built for high-performance embedded 3D display applications. Scalable Arm® Cortex®-A53 performance and embedded features, such as: multiscreen high-definition display support, 3D-graphics acceleration, 4K video acceleration, and extensive peripherals make the AM62Px well-designed for a broad range of automotive and industrial applications, including automotive digital instrumentation, automotive displays, industrial HMIs, and more.

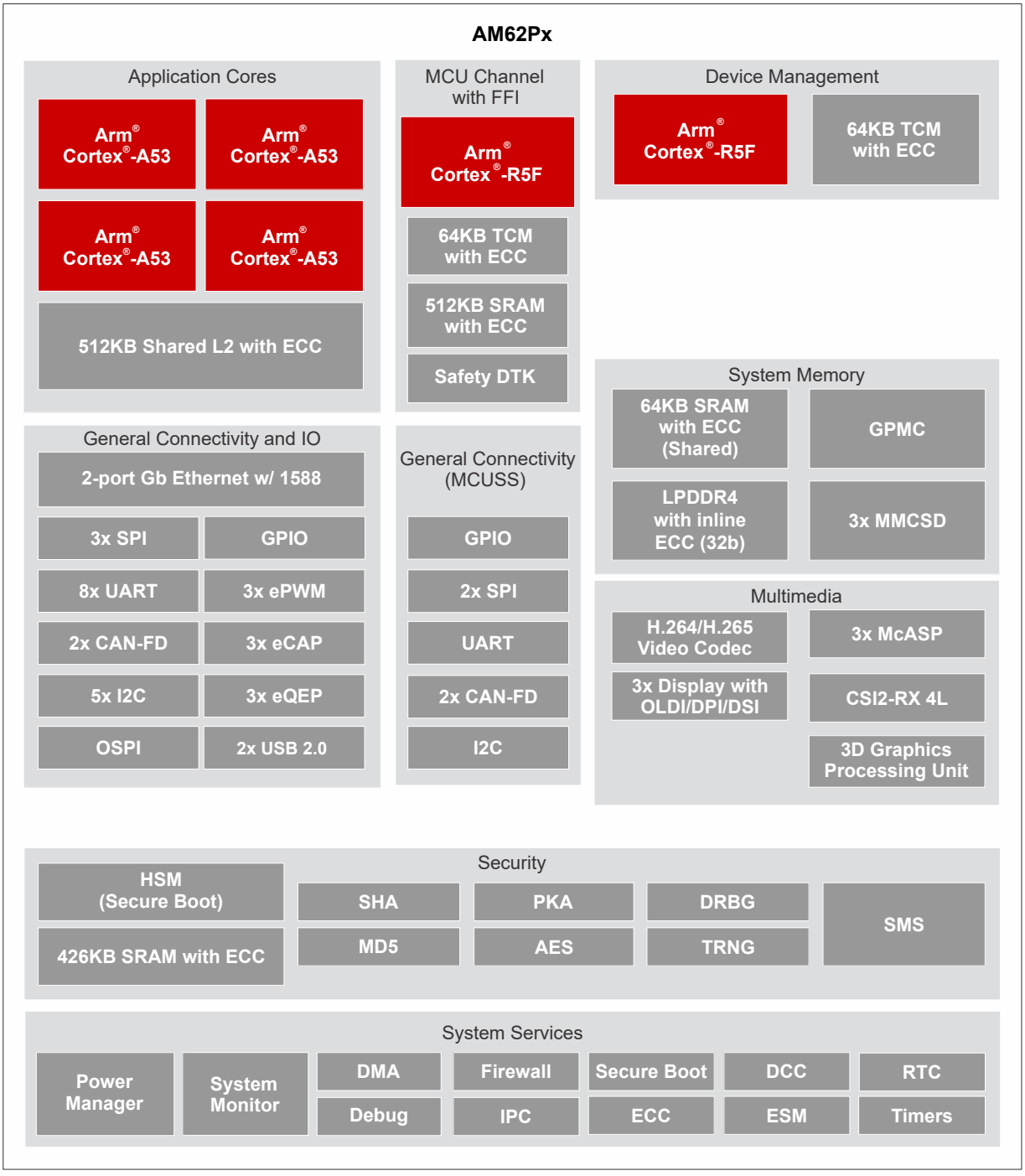

Figure 2-1. AM62P Block Diagram

2.1 Key Features and Benefits

AM62Px processors offer the following features and benefits:

- Focus on innovation and fast development with Linux® and Android™ SDK accompanied with real-time functional safety and security SDKs.
- Address the next wave of HMI designs with Triple Display over OLDI (LVDS) (1 × OLDI-DL, 1 × or 2 × OLDI-SL), DSI or DPI along with the new generation of 3D graphics processing unit (GPU) and 4K video acceleration.
- Provides intelligent features, such as facial recognition and touchless HMI with Arm Cortex-A53 central processing units (CPU) and open-source AI software and tools.

2.2 AM62P Display Subsystem

The AM62P DSS features the following:

- Triple display support over OLDI (LVDS) (1 × OLDI-DL, 1 × or 2 × OLDI-SL), DSI or DPI
  - OLDI-SL: up to 1920 × 1080 at 60 fps (165MHz pixel clock)
  - OLDI-DL: up to 3840 × 1080 at 60 fps (150MHz pixel clock)
  - MIPI® DSI: with 4-lane MIPI® D-PHY supports up to 3840 × 1080 at 60 fps (300MHz pixel clock)
  - DPI (24-bit RGB parallel interface): up to 1920 × 1080 at 60 fps (165MHz pixel clock)
- Four display pipelines with hardware overlay support. Use a maximum of two display pipelines per display.
- Supports safety features such as freeze-frame detection and data correctness check.

Figure 2-2 illustrates the DSS block diagram.

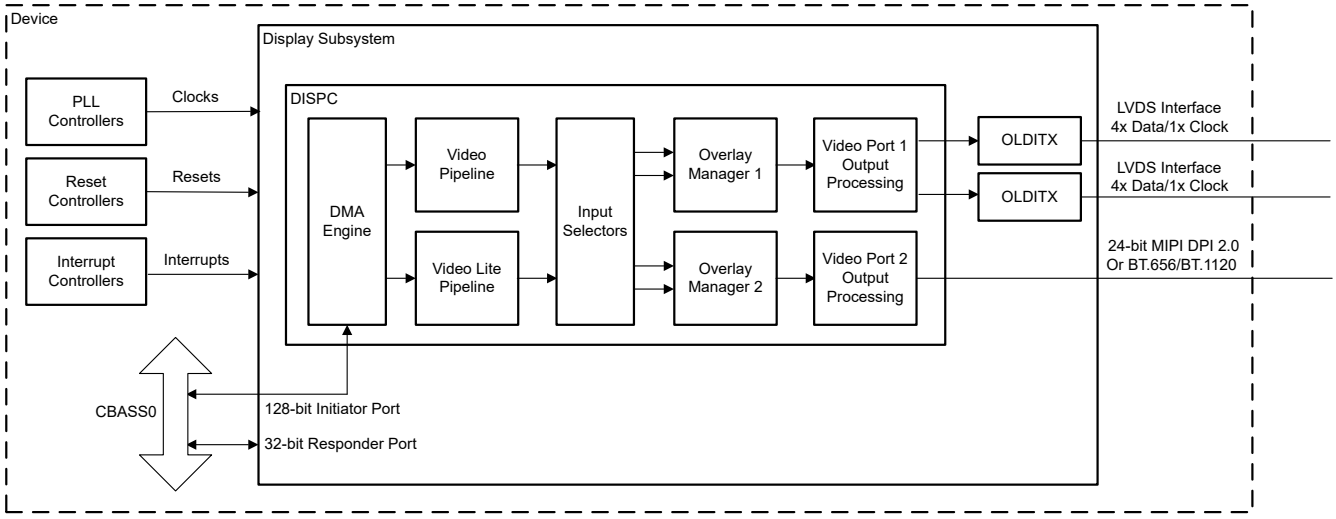

Figure 2-2. DSS Block Diagram

2.3 AM62P Display Subsystem Clocking Architecture

The AM62Px family of SoCs have two instances of the DSS controller (DSS0 and DSS1). These two DSS controllers are backed by three PLLs giving a possibility of a maximum of three independent streams of video data, namely OLDI, DPI, and DSI.

The two OLDIs can then be configured in the clone mode to have one pair of two cloned displays + two more independent displays giving a total output for four video sinks. There are two OLDI TXs available at the disposal (DSS0) of VP0 and a DPI output from VP1. DSS1 VP0 can only output to one of the two available OLDI TXs, allowing for a single-link mode. However, DSS1 VP0 can also bypass the OLDI TXs and simply provide a DPI output on the SoC.

DSS1 VP1 can also provide a DPI output. Along with that, the DSS1 VP1 can instead redirect the output to the DSI controllers and help send out DSI signals out of the SoC.

There are three VPs (DSS0-VP1, DSS1-VP0, DSS1-VP1) capable of driving DPI signals out of the SoC, but only one can do that at a given time. The DPI outputs of all these VPs are passed through a MUX and only a single set of DPI signals come out of the SoC.

PLL16 is the dedicated OLDI PLL and is only being used with DSS0-VP0. This PLL helps provide a maximum of 2.1GHz for dual-link mode operations.

PLL17 is the DPI PLL and provides pixel clock to all the three DPI output capable VPs, DSS0-VP1, DSS1-VP0, and DSS1-VP1. The maximum frequency is 165MHz. In a single case, PLL17 provides a 300MHz pixel frequency to DSS1-VP1 to drive a DSI output. This happens when the DSI PLL (PLL18) is being used to drive a single link output from DSS1-VP0.

PLL18 doubles as a DSI or OLDI clock. For DSI purposes, PLL18 provides a clock with a maximum frequency of 300MHz to DSS1-VP1. For the OLDI purposes, PLL18 supports the DSS1-VP0 for a single link OLDI output.

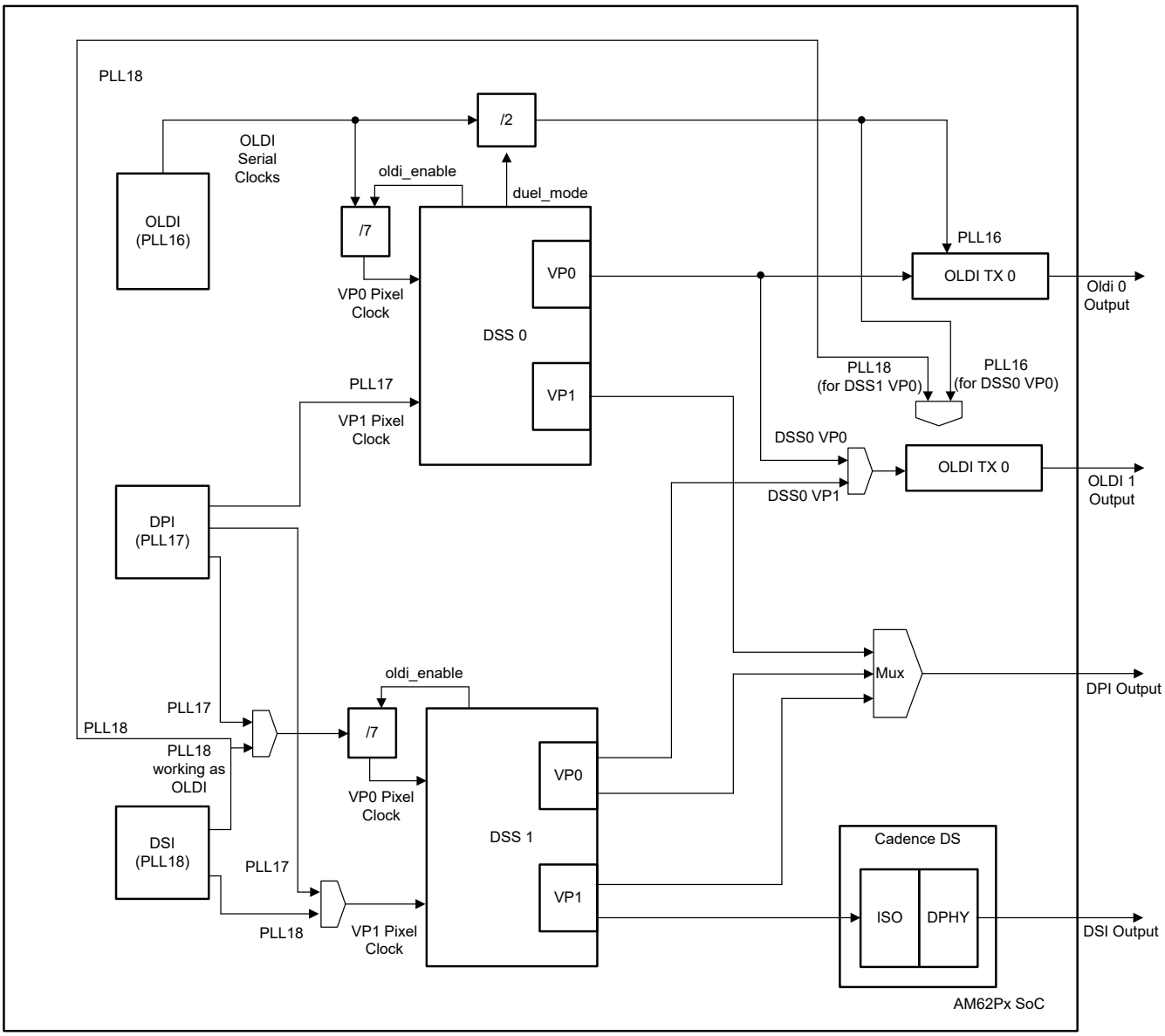

Figure 2-3. PLL18 Provides Clock Input for OLDIs TX in Dual-Link Mode or Single-Link Mode

PLL16 provides clock input for OLDI TXs in dual-link mode or single-link mode operation. Enabling `oldi_enable_cfg0` in DSS0 VP0 divides the input PLL16 clock by seven times as input to DSS0 VP0 pixel

clock. Enabling `dual_cfg0` in DSS0 VP0 divides the input PLL16 clock by two times for the PLL clock of OLDI TXs.

PLL18 provides clock input for OLDI TX1 in single link mode. Enabling oldi_enable_cfg0 in DSS1 VP0 divides the input PLL8 clock by 7 times as input to DSS1 VP0 pixel clock.

Figure 2-4 shows the screenshot of the DSS0 clock using the k3conf tool, also known as a K3conf clock dump snippet.

```
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 | 300000000  |
|   186   |     3    | DEV_DSS0_DPI_1_IN_CLK_PARENT_HSDIV0_16FFT_MAIN_17_HSDIVOUT0_CLK        | CLK_STATE_READY | 300000000  |
|   186   |     4    | DEV_DSS0_DPI_1_IN_CLK_PARENT_BOARD_0_VOUT0_EXTPCLKIN_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 | 320000000  |
|   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_DPI0_PCLK_OUT0     | CLK_STATE_READY | 300000000  |
|   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 | 300000000  |
root@am62pxx-evm:~#
```

Figure 2-4. DSS0 Clock Using K3 Configuration Tool

The following sections of the application note provide the requirements and steps to integrate the LVDS panel on Linux and the RTOS.

3 Display Signals and Timing Parameters

The *DIGI Embedded Documentation Portal* webpage provides definitions and an illustration of *LCD display signals and timing parameters*. These definitions are also valid for OLDI.

Every manufacturer provides display timings in a slightly different way and some provide more detail than others.

Once all the resolution and timing parameters are gathered, integrate them in either the Linux OS or RTOS using AM62P.

4 Steps for Integration (Linux®)

To support a display from an AM62P perspective, complete the following:

1. Update the device tree by defining a display overlay or modify the device tree directly with the display entry calling out the specific panel.
2. Add display timings to `panel-simple.c` with the same compatibility string.

SK-LCD1 support is enabled in the Linux driver. The code snippets in Section 4.1 and Section 4.2 show how to create a Linux overlay file and timing information using `panel-simple.c`, respectively.

For more information, see the following link:

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

4.1 Linux® Overlay File

The code snippet in this section is supported from the following reference:

- https://git.ti.com/cgit/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_oldi0_pins_default: main-oldi0-pins-default {
        pinctrl-single,pins = <
            AM62X_IOPAD(0x0260, PIN_OUTPUT, 0) /* (AA5) OLDI0_A0N */
            AM62X_IOPAD(0x025c, PIN_OUTPUT, 0) /* (Y6) OLDI0_A0P */
            AM62X_IOPAD(0x0268, PIN_OUTPUT, 0) /* (AD3) OLDI0_A1N */
            AM62X_IOPAD(0x0264, PIN_OUTPUT, 0) /* (AB4) OLDI0_A1P */
            AM62X_IOPAD(0x0270, PIN_OUTPUT, 0) /* (Y8) OLDI0_A2N */
            AM62X_IOPAD(0x026c, PIN_OUTPUT, 0) /* (AA8) OLDI0_A2P */
            AM62X_IOPAD(0x0278, PIN_OUTPUT, 0) /* (AB6) OLDI0_A3N */
            AM62X_IOPAD(0x0274, PIN_OUTPUT, 0) /* (AA7) OLDI0_A3P */
            AM62X_IOPAD(0x0280, PIN_OUTPUT, 0) /* (AC6) OLDI0_A4N */
            AM62X_IOPAD(0x027c, PIN_OUTPUT, 0) /* (AC5) OLDI0_A4P */
            AM62X_IOPAD(0x0288, PIN_OUTPUT, 0) /* (AE5) OLDI0_A5N */
            AM62X_IOPAD(0x0284, PIN_OUTPUT, 0) /* (AD6) OLDI0_A5P */
            AM62X_IOPAD(0x0290, PIN_OUTPUT, 0) /* (AE6) OLDI0_A6N */
            AM62X_IOPAD(0x028c, PIN_OUTPUT, 0) /* (AD7) OLDI0_A6P */
            AM62X_IOPAD(0x0298, PIN_OUTPUT, 0) /* (AD8) OLDI0_A7N */
            AM62X_IOPAD(0x0294, PIN_OUTPUT, 0) /* (AE7) OLDI0_A7P */
            AM62X_IOPAD(0x02a0, PIN_OUTPUT, 0) /* (AD4) OLDI0_CLK0N */
            AM62X_IOPAD(0x029c, PIN_OUTPUT, 0) /* (AE3) OLDI0_CLK0P */
            AM62X_IOPAD(0x02a8, PIN_OUTPUT, 0) /* (AE4) OLDI0_CLK1N */
            AM62X_IOPAD(0x02a4, PIN_OUTPUT, 0) /* (AD5) OLDI0_CLK1P */
        >;
    };
};

&dss {
    pinctrl-names = "default";
    pinctrl-0 = <&main_oldi0_pins_default &main_dss0_pins_default>;
};

&dss_ports {
    #address-cells = <1>;
    #size-cells = <0>;

    /* VP1: LVDS Output (OLDI TX 0) */
    port@0 {
        reg = <0>;
        oldi_out0: endpoint {
            remote-endpoint = <&lcd_in0>;
        };
    };

    /* VP1: LVDS Output (OLDI TX 1) */
    port@2 {
        reg = <2>;
```

```
        oldi_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 Timing Information

The code snippet in this section is supported from the following reference:

- https://git.ti.com/cgit/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 Steps for Integration (RTOS)

Integrate a new LVDS panel in RTOS using this SysConfig configurator which is part of the MCU_PLUS_SDK. The sysconfig for the DSS driver provides the following configuration tabs to integrate a new LVDS panel.

1. Video Port Timing Configuration:
   - Configure the *Pixel Clock Frequency* required for the panel
   - Timing parameters HSYNC, VSYNC, VFP, VBP, HFP, HBP and more
   - Interface width: 24 bit or 18 bit
   - Timing signals polarity
2. OLDI Configuration:
   - Map type for OLDI: VESA, JEIDA, 24-bit or 18-bit standards
   - Input bit width: Whether panel connected expects 24-bit or 18-bit input
   - Dual link mode: Enable or disable dual-link mode
   - Data enable signal polarity

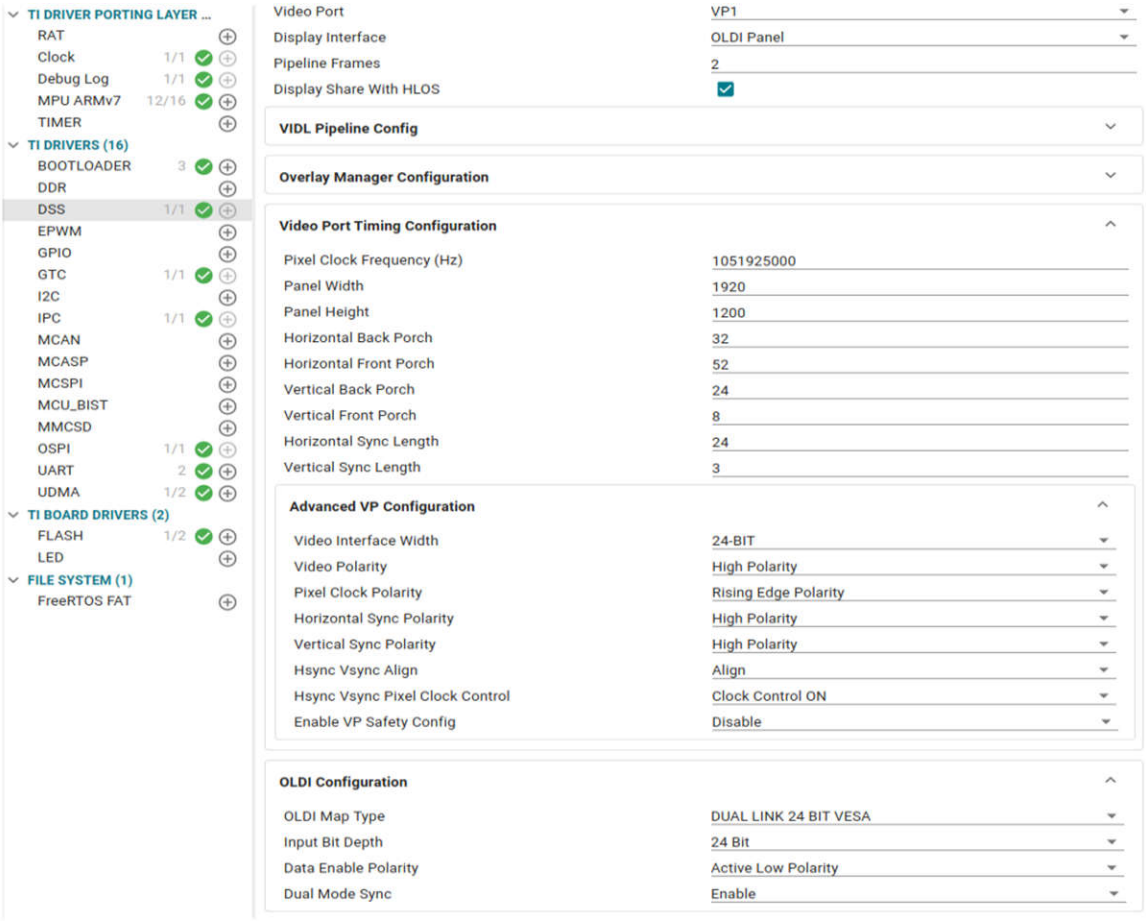

Figure 5-1. SysConfig Configurator

6 Terminology

LVDS	Low-Voltage Differential Signaling
OLDI	Open LVDS Display Interface
RTOS	Real-Time Operating System
SoC	system-on-a-chip
HMI	Human Machine Interfaces
SDK	Software Development Kit
OLDI-SL	OLDI-Single Link
OLDI-DL	OLDI-Dual Link
DSI	Display Serial Interface
DPI	Display Parallel Interface
MIPI	Mobil Industry Processor Interface
RGB	red, green, blue
DSS	Display Subsystem
PLL	Phase-Locked Loop
HSYNC	Horizontal synchronization
VSYNC	Vertical synchronization
DRDY	Data Ready, Data Enable
DE	Data Enable
VBP	Vertical Back Porch
VFP	Vertical Front Porch
HBP	Horizontal Back Porch
HFP	Horizontal Front Porch
OS	Operating System

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265