

# Adding LVDS Support to Panda 5 Board

Runhong Deng, Reno Ramsey

## 1. Configuration of Demo System

In the LVDS demo system, Texas Instruments DS90UH925 serializer is used to transfer display data from Panda 5 board to LVDS (FPD-Link III) signal. DS90UH928 deserializer is used to transfer LVDS to Open LDI and drive a 10 inch display. Besides carrying display signal, a bidirectional control channel (BCC) can provide communication between serializer/deserializer as well as remote I2C slave devices. In the setup, BCC allows OMAP 5 to talk to Atmel touch controller (mxt768E).

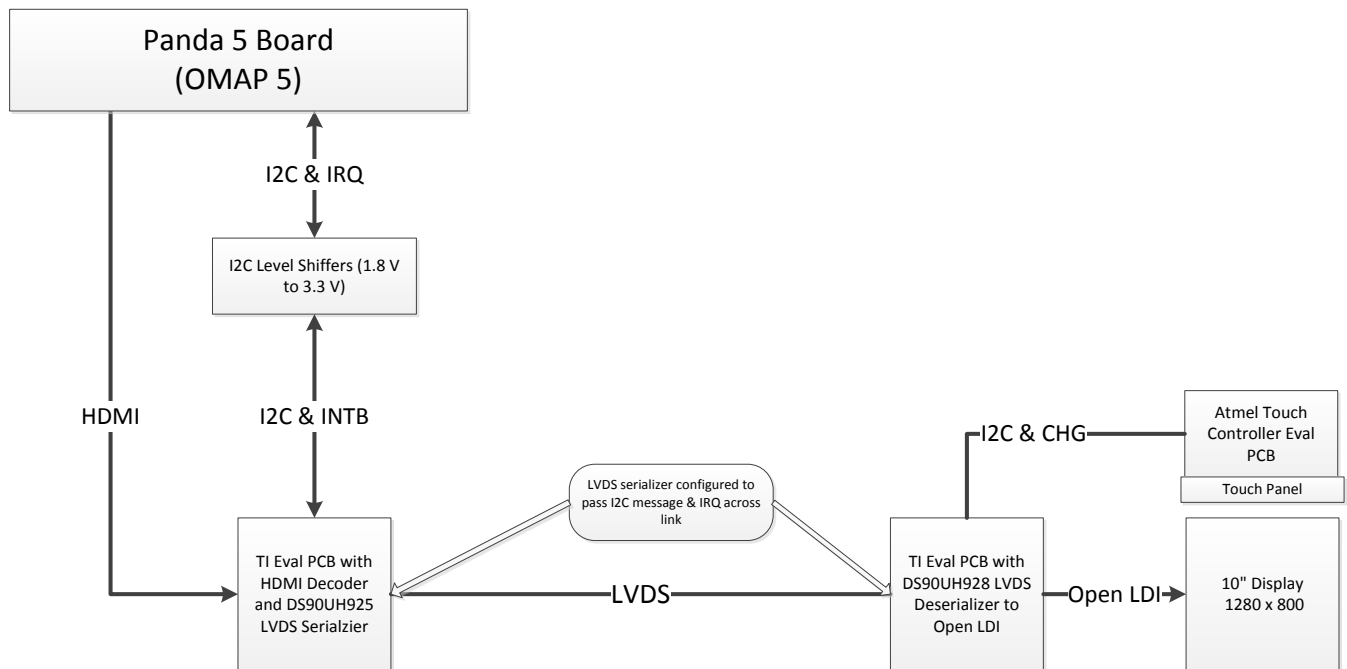


Fig 1. System Block Diagram

Function block diagram of DS90UH925 and DS90UH928 are shown in Figure 2 and 3, respectively.

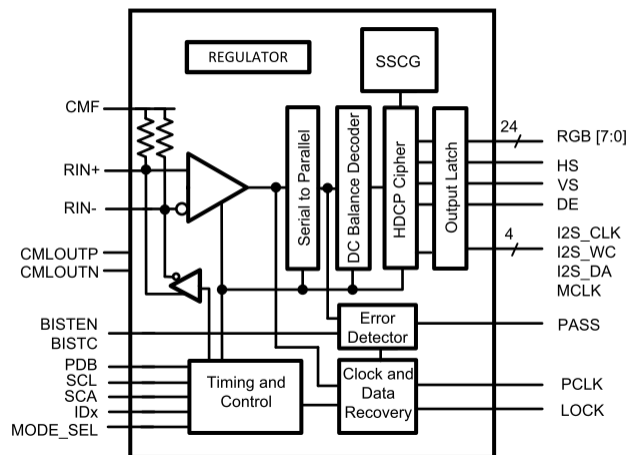


Fig 2. Function Block Diagram of DS90UH925 (Copied from Datasheet)

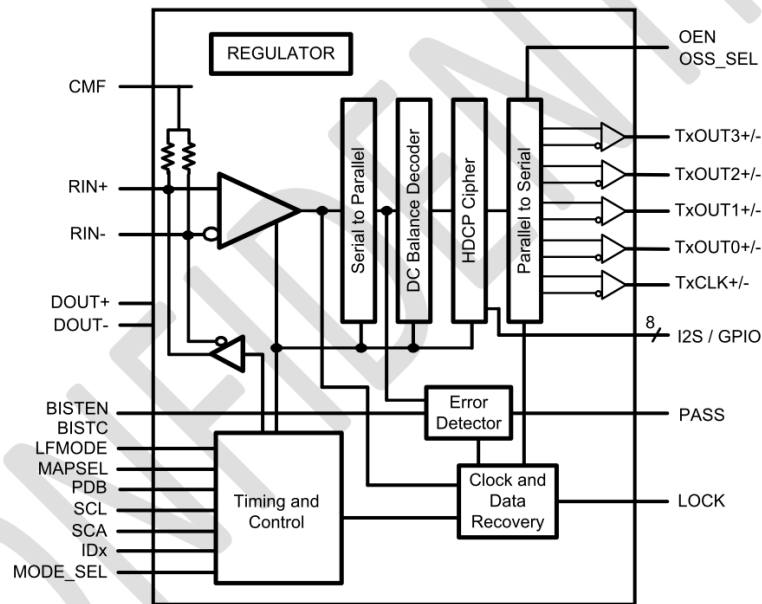


Fig 3. Function Block Diagram of DS90UH928 (Copied from Datasheet)

On DS90UH925 evaluation board, a HDMI receiver, Analog Devices ADV7611, is used to transfer HDMI signal to 24bit RGB signals. The block diagram of ADV7611 is shown in Figure 4.

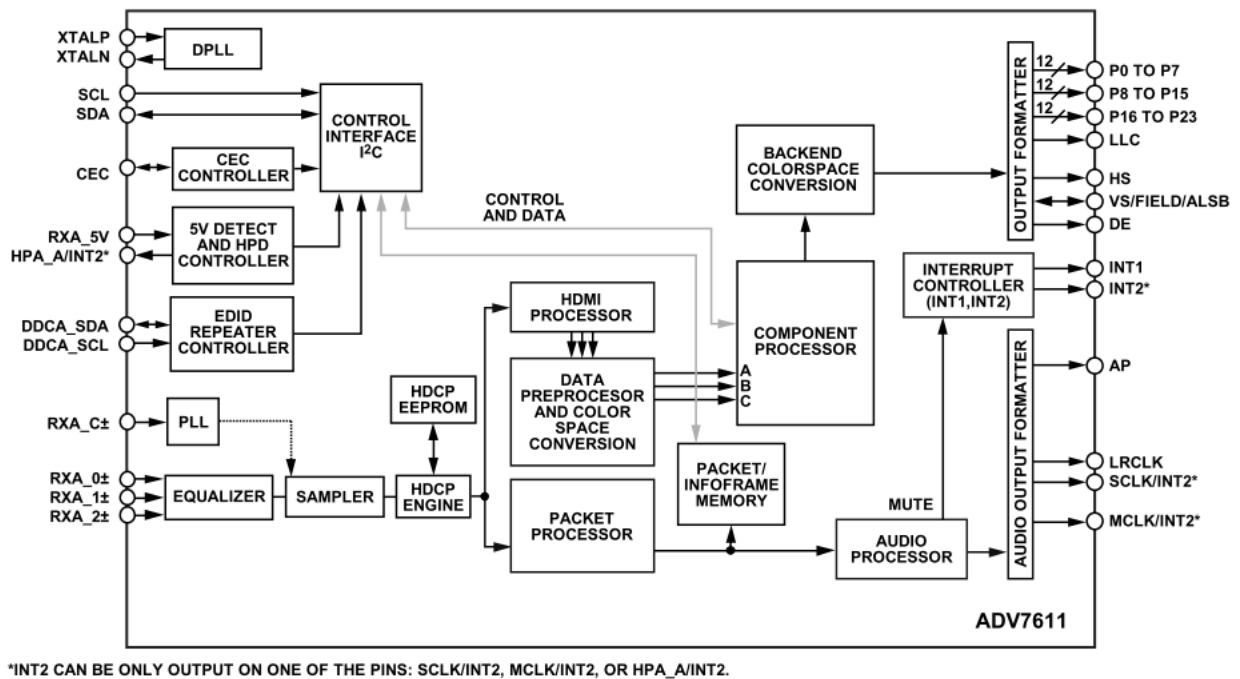


Fig 4. Function Block Diagram of ADV7611 (Copied from Datasheet)

A TI microcontroller, MSP430 is used in DS90UH925 evaluation board for configuring ADV7611 and DS90UH925. The microcontroller is removed in the demo system since OMAP5 takes care of all the configuration.

The I2C bus and GPIO from DS90UH925 evaluation board are at 3.3V level. Panda 5 board accepts 1.8V in J17 (I2C bus and GPIO). A NXP PCA9306 chip is used as I2C level shifter.

## 2. Software Modification

The operation system is Android 4.2.2 Source code is available from TI. The link is below –

[http://omapedia.org/wiki/Panda5AJ.1.5\\_Release\\_Notes](http://omapedia.org/wiki/Panda5AJ.1.5_Release_Notes)

Please check the documents below for details on adding support for Atmel touch controller and for different LCD panel.

*Modify Android to Support Atmel Touch Controller, R. Deng*

*Modify Android to Support Different LCD Monitor, R. Deng*

The following code is to add I2C support for DS90UH925 and ADV7611.

```
static struct i2c_driver lvdsi2c_driver = {
    .driver = {
        .name = "lvdsi2c",
    },
    .probe = lvdsi2c_probe,
    .remove = lvdsi2c_remove,
    .id_table = lvdsi2c_id,
};
module_i2c_driver(lvdsi2c_driver);

static struct i2c_board_info __initdata panda5_i2c_2_boardinfo[] = {
    {
        I2C_BOARD_INFO("atmel_mxt_ts", 0x4d),
        .platform_data = &mxt_platform_data,
    },
    {
        I2C_BOARD_INFO("lvdsi2c", 0x14),
    },
};
```

Since MSP430 is removed, initialization of DS90UH925 and ADV7611 needs to be done by OMAP5. The initialization procedure is listed in the [LVDS-DemoSetUp.xlsx](#).

EDID data of the LCD panel should be read. The EDID data is stored into a new file 'edid\_data.h'. During initialization, OMAP5 send the EDID data to ADV7611 through I2C bus.

For the touch function, CHG pin from Atmel touch controller turns LOW when a touch is sensed. The information is passed back to DS90UH925 and forced an INTB pin from DS90UH925 to LOW. After OMAP5 read messages from touch controller. A read to DS90UH925 HDCP\_ISR register is needed in order to clear the interrupt at DS90UH925, releasing INTB. This is done in atmel\_mxt\_ts.c.

```
static irqreturn_t mxt_interrupt(int irq, void *dev_id)
```

```

{
    struct mxt_data *data = dev_id;
    union i2c_smbus_data value;

    value.byte = 0;
    i2c_smbus_xfer(data->client->adapter, 0x14, 0, I2C_SMBUS_READ, 0xc7, I2C_SMBUS_BYTE_DATA,
    &value);
    ...

```

### 3. Acknowledge

Ian Byers from TI provided guidance for implementation, especially the initialization of DS90UH925 and ADV761.

### 4. Revision

Date	Author	Notes
06/10/2013	R. Deng, R. Ramsey	Initial release