

Connecting Video Output Devices to PNX1500 and PNX1700

NDK 6.1

Application Note v0.5, 24 June 2009

Overview

This application note provides details about connecting video encoders to the PNX1500 and PNX1700 series of Nexperia Media Processors. This application note provides details about connecting video encoders to the PNX1500 and PNX1700 series of Nexperia Media Processors. It does not cover the PNX1005. A separate document, [PNX1005 Interface Reference Manual](#) covers the PNX1005. The background provided here will also prove helpful to designers who wish to connect up a video encoder to be driven by the Nexperia Media Processor. If the rules and conventions here are followed, the whole software stack will work to its full capabilities and without modifications.

No matter what Digital Encoder (Denc)¹ you choose, at the very least, the following must be considered:

- What pins are connected between the Nexperia Media Processor and the Denc?
- What software setup must happen on the Nexperia to match this?
- What registers must be setup inside of the Denc to match this mode?
- What other board specific hardware might need to be configured and where should the code necessary to do this be located?

Software Drivers

It is a basic premise of the video software used in Nexperia Media Processors that a Board Support Library (BSL) is provided with the system so as to setup and control the video hardware in a standard way. When a proper BSL is provided, all of the standard Nexperia programs can use high level interfaces to address the video output function without changes. The BSL provides function tables that are used by the higher-level software layers. The interface used to control the video output hardware is tmVencAna. The VencAna interface controls the analog video encoder. This concept is elaborated in the *Video I/O APIs* volume. See [Chapter 1 Video Architecture Overview](#) and [Chapter 11 Analog Video Output BSL: tmVencAna](#).

1. The term “denc” is often used here to mean a device that is traditionally a D/A converter for video. In the latest systems supporting DVI or HDMI, the “denc” may actually convert the raw digital signal into a digital signal suitable for transmission to the outside world. But we still call it a “denc”.

Board Support Library Examples

Several example Board Support Libraries are available. Depending on what you wish to accomplish, you should start from the example closest to your needs. This table helps you understand the differences when it comes to video output.

Board	What it demonstrates
mdsbslDmaThin	Uses a 7104 only to support NTSC, PAL and YPbPr component video up to 1080i resolution.
mdsbslDma	Uses a 7104 to support NTSC, PAL and component output. Also uses a TFP410 to support a DVI output. Properly announces this support so that exolVideoSimple can find it.
mdsbslLcp	Uses three optional daughter boards to support NTSC and PAL (7104), VGA (7104) and/or DVI (TFP410) outputs. Unlike the DMA, does not support switching between NTSC and VGA or DVI without reconfiguring the hardware.
tmbslNRef	Also uses the 7104 for NTSC and VGA output. Has support for DVI. This is the oldest of the BSLs and it is not up to date with the latest specs. Its continued operation proves the backwards compatibility of the latest changes.

1 Analog Video Output with at Denc

Since the SAA7104 Digital Encoder (Denc) and its variant, SAA7105, are often used for video output with Nexperia Media Processors, this application note refers to systems that use these chips.

1.1 Hardware Connections

The following table lists the tested and recommended video signal connection between the Nexperia Media Processor and the SAA7104/5.

Table 1: Recommended Video Signal Connections

Nexperia Media Processor Internal Signal Name (PNX: VDO_CLK1=1 / VDO_CLK1=0) (7104/5: falling edge / rising edge)	PNX15XX/PNX17XX pin name	SAA7104/5 pin name
QVCP_CLK	VDO_CLK1	PIXCLKI
AC bypass to ground on SAA7104/5	(none)	PIXCLKI
QVCP_HSYNC	VDO_D[30]	HSVGC
QVCP_VSYNC	VDO_D[29]	VSVGC
G[3] / R[7]	VDO_D[16]	PD11
G[2] / R[6]	VDO_D[15]	PD10
G[1] / R[5]	VDO_D[14]	PD9
G[0] / R[4]	VDO_D[13]	PD8
B[7] / R[3]	VDO_D[12]	PD7
B[6] / R[2]	VDO_D[11]	PD6
B[5] / R[1]	VDO_D[10]	PD5
B[4] / R[0]	VDO_D[9]	PD4
B[3] / G[7]	VDO_D[8]	PD3

Table 1: Recommended Video Signal Connections

Nexperia Media Processor Internal Signal Name (PNX: VDO_CLK1=1 / VDO_CLK1=0) (7104/5: falling edge / rising edge)	PNX15XX/PNX17XX pin name	SAA7104/5 pin name
B[2] / G[6]	VDO_D[7]	PD2
B[1] / G[5]	VDO_D[6]	PD1
B[0] / G[4]	VDO_D[5]	PDO

1.1.1 A Note about Pixel Clock

The Denc typically needs to have a pixel clock. This clock should be generated by Nexperia's programmable clock output controlled with the QVCP and tmVrendGfxVo. To the extent possible, this clock should be used for all operations of the Denc. Using a local crystal for some operations of the Denc and the programmable clock for the pixel clock can lead to subtle or not so subtle artifacts in the video. It is normal for the Nexperia to be the master of the pixel clock. If you have a reason to make it the slave to the pixel clock, you should carefully consider the impact this will have on the overall software system.

1.2 BSL Support

When selecting this hardware connection, the BSL must include the necessary code to drive this connection. Let us first examine the example BSLs to see how this is done.

1.2.1 Selecting Appropriate Settings for the Chosen Configuration

The user's application chooses the requested video mode by selecting an adapter type (output jack) and a video mode. Then the function `applyExtInterfaceSettings()` is called to place the hardware in this mode. Refer to the `DmaThin` BSL. This board uses a 7104 to provide NTSC, PAL and component (YPbPr) video support. The BSL table entry for `applyExtInterfaceSettings()` is populated with the function `avoApplyExtInterfaceSettings()`. Looking into this function, we see a switch statement that bases its action on the selected encoding type. It could also do this based on the adapter type and video mode. But we know that the encoding type has enough information here. The code must set the video output router (VDO router) to the right mode:

```
tmhwVdiVdoRouter_Ensure0 (); // ensure that the VdiVdo Router module is ready
tmhwVdiVdoRouter_SetVideoOutputMode(...); // set the router for this mode.
```

Note that the router mode is different for different video types. It is `tmhwVdiVdoRouter_VideoOutput24BitYuv0rRgb` for CVBS or SVideo outputs. It is `tmhwVdiVdoRouter_VideoOutput24BitYuv0nBothClkEdges` for VGA or YPbPr component video. `VideoOutput24BitYuv0nBothClkEdges` actually maps to `VDO_MODE = 0x44`. It is better to use the `VdiVdo` router library.

The `ApplyExtInterfaceSettings()` function must put the pixel clock in the right mode. This is somewhat tricky with the 7104. The SAA7104/5 can accept a differential clock on `PIXCLKI` and `/PIXCLKI` though in a typical Nexperia Media Processor application it is driven with a single-ended clock. Due to some particularities in the chip implementation, the unused `/PIXCLKI` pin should be connected to ground through a 22 pF or larger capacitor. This will prevent noise or RFI from interfering with the chip operation.

When outputting VGA or YUV video:

- Set the pixel clock mode to single ended by setting bit 7 in the SAA7104/5 sub-address register 0x84.

When outputting CVBS or S-Video:

- Set the pixel clock mode to differential by clearing bit 7 in SAA7104/5 sub-address register 0x84.

Notice how the code reads the existing value (over I2C) and then writes back the modified value. This is a good place to point out that most of the configuration of a Denc like this typically happens over I2C. The correct library to use is `tmdlI2C`, though existing code does not always follow this recommendation. Because the I2C communication requires interrupts to occur, it is not a good idea to reconfigure the Denc from inside of an interrupt.

1.2.2 Denc (SAA 7104/5) Input Format

A Denc like the SAA7104/5 can operate in multiple input modes. To match this configuration, it should operate using Input Format mode 0. This is normally set by the `tmbsl7104open()` call and need not be programmed explicitly by the user application, e.g. `tmbslsaa7104.c`.

```
switch ( pSetup->encodingType )
{
case tmbslVencAna_Vga: /* RGB & HV syncs */
    saa7104SetReg(avoUnit, 0x3a, 0x29); // no embedded sync
    saa7104SetReg(avoUnit, 0x54, 0x01); // vsm pin provides v-sync
    saa7104SetReg(avoUnit, 0xDC, 0x05); // HD sync engine disabled
    saa7104SetReg(avoUnit, 0xFD, 0xC3); // mode 0
    break;
```

1.2.3 Boards with Multiple Dencs

Several of the reference boards have included more than one Denc. The `mdsbslDma` example BSL is presently the best example of how to do this. Some of the history of the interface is visible here thus it requires a bit more explanation.

Refer to the function `avoChangeComponentBinding()`. This is registered in the `tmbslVencAnaExt` interface table. It is called when the chosen adapter type is selected. It repopulates the `VencAna` BSL tables with functions appropriate to the Denc connected to this jack.

Refer to the function `avoGetSuppAdapterTypes()`. This is registered in the `tmbslVencAna` interface. Since the DVI support is optional, it checks whether the DVI hardware is present. Then it fills in the list of supported adapters appropriately.

2 Driving TFT LCD Panel

TFT LCD Panels come in a variety of different pinouts. The following is a summary of all the necessary information.

2.1 Video Output Router Mode

The Nexperia Media Processor is a System-On-A-Chip (SoC), hence certain signals are managed at the system level.

- At the peripheral level there are FGPO and QVCP.
- At the chip level there are 40 `VDO_D*` pins that are associated with FGPO and QVCP.

The Video Output Router is a system-level control that determines how the FGPO and QVCP peripheral are connected to the pins. FGPO, when it is used, and QVCP must be properly configured to generate the correct signals to be sent to the Video Output Router which will route the selected signal to the pins.

For TFT LCD operation, the Video Output Router Mode 0 should be selected to generate TFT LCD signals. This mode is described in both the PNX15xx Series and PNX17xx Series Databooks, in Chapter 3: *System on Chip Resources*, section 7.1 *MMIO Registers for the Input/Output Video/Data Router*, Table 8 *Global Registers*.

The relevant information is duplicated here for clarity.

Bit	Symbol	Access	Value	Description
2:0	VDO_MODE	R/W	0	TFT/QVCP mapping to VDO interface 000*: TFT LCD controller with 24- or 18-bit digital RGB/YUV video TFT_DATA[23:0] -> VDO_D[28:5] TFT_VSYNC -> VDO_D[29] TFT_HSYNC -> VDO_D[30] TFT_DE -> VDO_D[31] TFT_VDDON -> VDO_D[4] TFT_BKLTON -> VDO_D[3] TFT_CLK -> VDO_CLK1 In 18-bit mode VDO_D[28:23] -> R[5:0] or Y[5:0] VDO_D[20:15] -> G[5:0] or U[5:0] VDO_D[12:7] -> B[5:0] or V[5:0] In 24-bit mode VDO_D[28:21] -> R[7:0] or Y[5:0] VDO_D[20:13] -> G[7:0] or U[5:0] VDO_D[12:5] -> B[7:0] or V[5:0]

2.2 Nexperia Media Processor QVCP To LCD Connection

This table lists the internal TFT LCD signal names and the corresponding VDO_D* pin names for the TFT LCD interface connection.

Table 2: Internal TFT LCD Interface Connection Information

Nexperia Media Processor Internal Signal Name	Pin Name for 24-bit Mode	Pin Name for 18-bit Mode
TFT_DE	VDO_D[31]	VDO_D[31]
TFT_VDDON	VDO_D[4]	VDO_D[4]
TFT_BKLTON	VDO_D[3]	VDO_D[3]
TFT_CLK	VDO_CLK1	VDO_CLK1
TFT_HSYNC	VDO_D[30]	VDO_D[30]
TFT_VSYNC	VDO_D[29]	VDO_D[29]
R[7] or Y[7]	VDO_D[28]	
R[6] or Y[6]	VDO_D[27]	
R[5] or Y[5]	VDO_D[26]	VDO_D[28]
R[4] or Y[4]	VDO_D[25]	VDO_D[27]
R[3] or Y[3]	VDO_D[24]	VDO_D[26]
R[2] or Y[2]	VDO_D[23]	VDO_D[25]
R[1] or Y[1]	VDO_D[22]	VDO_D[24]
R[0] or Y[0]	VDO_D[21]	VDO_D[23]
G[7] or U[7]	VDO_D[20]	
G[6] or U[6]	VDO_D[19]	
G[5] or U[5]	VDO_D[18]	VDO_D[20]

Table 2: Internal TFT LCD Interface Connection Information

Nexperia Media Processor Internal Signal Name	Pin Name for 24-bit Mode	Pin Name for 18-bit Mode
G[4] or U[4]	VDO_D[17]	VDO_D[19]
G[3] or U[3]	VDO_D[16]	VDO_D[18]
G[2] or U[2]	VDO_D[15]	VDO_D[17]
G[1] or U[1]	VDO_D[14]	VDO_D[16]
G[0] or U[0]	VDO_D[13]	VDO_D[15]
B[7] or V[7]	VDO_D[12]	
B[6] or V[6]	VDO_D[11]	
B[5] or V[5]	VDO_D[10]	VDO_D[12]
B[4] or V[4]	VDO_D[9]	VDO_D[11]
B[3] or V[3]	VDO_D[8]	VDO_D[10]
B[2] or V[2]	VDO_D[7]	VDO_D[9]
B[1] or V[1]	VDO_D[6]	VDO_D[8]
B[0] or V[0]	VDO_D[5]	VDO_D[7]

Note: Some LCDs appear to label DE as composite sync.

3 DVI Output

The DMA BSL, `mdsblDma`, supports the TFP410 Denc for DVI output. A BSL specific to this part also exists. It is `mdsblTfp410`. Using this, DVI output is relatively simple. The VDO mode is set to `tmhwVdiVdoRouter_VideoOutput24BitYuv0rRgb`.

4 HDMI Output

The PNX1005 demo board includes a fully implemented HDMI output driver based on the TDA9983. Porting this code back to the PNX1500 or PNX1700 is left as an exercise for the student. You can always contact <http://www.TCSHelp.com> for more information.