Application Report
Mar 2012

TMS320DM814X Boot Logo on external device

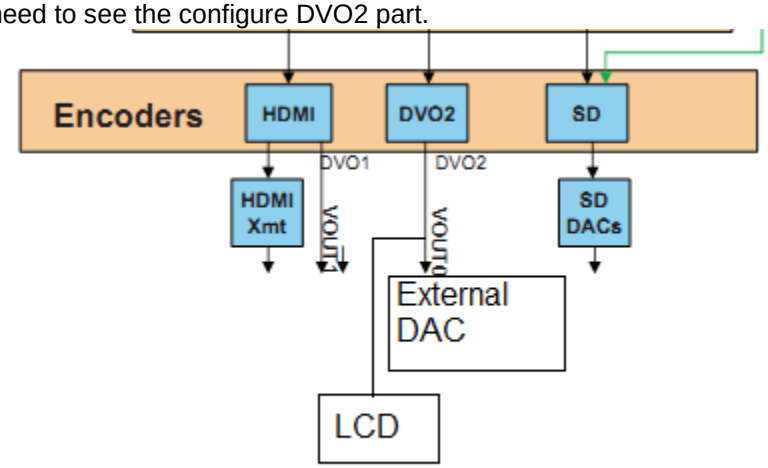
Video Surveillance Applications

ABSTRACT

This application note descript that how to implement the boot logo on external device

1. Overview

Here we use external DAC device THS8200 as example for how to make uboot work on 814x platform.
We tied the PLL of HDMI and DVO2, so they output the same timing and same content. For LCD user, you only need to see the configure DVO2 part.



2. Steps to make boot logo works on external device

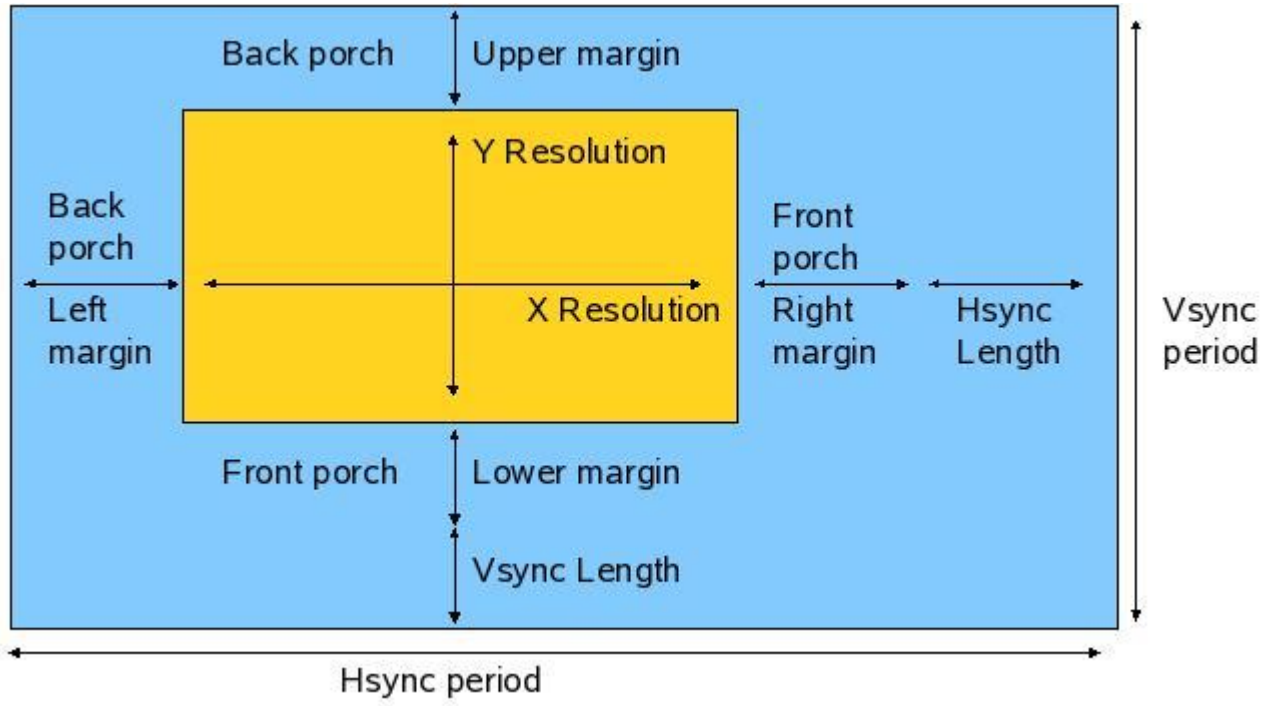
2.1. Configure DVO2 output timing

The output timing is configured in ti814x_set_mode, because we tied the HDMI and DVO2 timing, so if you want change the output timing, you should do following

```
ti814x_pll_config_hdmi(27000000);
ti814x_hdmi_enable(27000000);
```

Here you can change the 27000000 to the frequency you want.
And also you need change the output display parameter with function
ti814x_vps_configure_venc(uint32_t cfg_reg_base, int hdisp, int hsyncstart, int hsyncend, int htotal, int vdisp, int vsyncstart, int vsyncend, int vtotal, int enable_invert, int hs_invert, int vs_invert,char *cmd)
Here cfg_reg_base should be the HD_VENC_D_VOUT0(0xA000),

hdisp is X resolution,
hsync start is h back porch + X resolution + h front porch,
hsyncend is h back porch + X resolution + h front porch + h sync length
htotal is h sync period, but be aware, the htotal is not the same as hsyncend, it should be htotal >= hsyncend
vdisp is Y resolution,
vsync start is v back porch + Y resolution + v front porch,
vsyncend is v back porch + Y resolution + v front porch + v sync length
vtotal is v sync period, but be aware, the vtotal is not the same as vsyncend, it should be vtotal >= vsyncend
the enable_invert is enable the invert of VOUT_ACTIVEID signal
the hs_invert is to invert h sync output signal.
The vs_invert is to invert the v sync output signal.
The cmd is to decide if you want display the internal test signal “colorbar” or real bmp logo.



2.2. Configure external device

To configure the external device, you need to know which I2C bus the device is connected to.
For device THS8200, we use function “do_i2c_write_ths8200” to configure the device.
First set the I2C bus to correct one using i2c_set_bus_num.
Configure the device to correct mode. Now we using 10bit RGB separate sync mode, output 1920x1080@60Hz, you can change the “static uchar cfgTab[]” to make different output mode.
PS: current code using continuous write mode instead of byte by byte write. So the cfgTab register should be continuous. Otherwise you need change the i2c write let this “i2c_write(chip, addr,1, buffer,1)” and repeat this to finished all the configurations. But beware of that, in THS8200 configuration, it cost about several seconds to finish write all the register byte by byte. In continuous mode, it only cost about 130ms.

2.3. Configure uboot

You need put bmp in memory space like this
Tftp 0x81000000 logo.bmp
In the example, we flash the logo.bmp to nand flash address 0xA00000, and set bootcmd to

"nand read 0x81000000 0xa00000 0x600000;i2c ths8200;logo on 0x81000000 36000 60;nand read 0x81000000 0x00280000 0x300000;bootm 0x81000000"

Here "i2c ths8200 configure the external DAC device and "logo on 0x81000000 36000 60" let the logo on.