DaVinci PSP 03.01 Linux Installation User Guide

Link: → DaVinci PSP 03.01 Linux Installation User Guide

DaVinci PSP Overview

This is the installation guide for DaVinci PSP 03.01 - open source community based PSP release. DaVinci PSP 03.01 release supports DaVinci DM644x, DM365 and DM355 SoCs and the corresponding Evaluation Modules (EVM). The PSP release includes following components:

- DaVinci Linux Kernel with the support for DM365 and DM355 devices (baseport and device drivers)
  - Phase 1 Release Linux Kernel Baseline Version 2.6.31-rc5
  - Phase 2 and Phase 3 Releases Linux Kernel Baseline Version 2.6.32-rc1
- Arago filesystem version 2009.09. More information on Arago file system available at the Arago website [1]
- Secondary bootloader - U-Boot 2009.06-rc0 with the support for DM355 device
- Primary bootloader - User Boot Loader (UBL) for NAND boot mode
- Serial flash utilities and CCS based NAND Writer utilities
- The Pre-built binaries for Kernel, Filesystem and U-Boot are compiled using CodeSourcery GNU Toolchain for ARM Processors
  - Phase 1 Release compiled with 2007q3-51 version [2]
  - Phase 2 and Phase 3 Releases compiled with 2009q1-203 version [3]
- The Pre-built binaries for UBL and NAND Writer utilities are compiled using Code Composer Studio (CCS) 3.3.38 installed on Windows XP with Service Pack 2
- The Pre-built binary for Serial flash utility is compiled using CodeSourcery GNU Toolchain and Mono Framwork [4]

Downloading the Release

Please refer to the **DaVinci GIT Linux Kernel Releases** for the latest release.

System Requirements

Building and running all of the PSP components requires both a Windows XP and a Linux machine.

The Windows XP (with Service Pack 2) machine is required for running CCS 3.3.38. CCS is required for building the UBL and NAND Writer utilities. NAND Writer should be executed in CCS environment to program UBL and U-Boot binaries on to the NAND device on the EVM.

Linux host is required for compiling U-Boot, Linux kernel and filesystem in the Arago Build environment. Linux host can also be used to host the TFTP server (for downloading images) and NFS server (for the root filesystem).

DaVinci PSP 03.01 Installation on DM355 EVM

DaVinci PSP Installation on DM355 EVM covers how to install the DaVinci PSP 03.01 release software components on DM355 EVM.

DaVinci PSP 03.01 Installation on DM36x EVM

DaVinci PSP Installation on DM36x EVM covers how to install the DaVinci PSP 03.01 release software components on DM36x EVM.

DaVinci PSP 03.01 Installation on DM644x EVM

DaVinci PSP Installation on DM644x EVM covers how to install the DaVinci PSP 03.01 release software components on DM644x EVM.

Migration to DaVinci PSP 03.01

This section provides details on the migration path for customers using Montavista Professional Edition 5.0 based LSP releases (LSP 2.xx releases).

Montavista Pro 5.0 based LSP 2.00 release used the kernel version 2.6.18. DaVinci PSP 03.01 is the first step to get closer to the Open Source Kernel version (kernel.org). DaVinci PSP 03.01 release uses 2.6.31-rc2 Kernel or above.

There are generic changes in the core part of the kernel in 2.6.31-rc2 or above, compared to 2.6.18. The baseport (architecture) for DaVinci SoC's have been re-arranged after reviewing the architecture with ARM sub-system maintainers. Other changes are mainly in the sub-system specific core code (e.g. V4L2, USB, MMC/SD, ALSA/Sound). For e.g. V4L2 sub-system introduced the framework called sub-device model to be used as an interface between Capture driver and the decoder drivers. MMC/SD sub-system improved the performance using a better buffer allocation/management mechanism and also brought in SDIO support. Details on the DaVinci specific device drivers are given below.

Baseport SoC, EVM support

DaVinci PSP 03.01 release re-arranged the DaVinci architecture (arch/arm/mach-davinci) to clearly separate SoC specific and EVM/Board specific details, with the goal to re-use the common drivers across multiple SoCs. The baseport has a better pinmux, interrupt and EDMA event mux capabilities and better clock structures. The header files have also been re-arranged and moved to arch/arm/mach-davinci/include/mach folder (used to be in include/asm-arm/arch-davinci folder).

EDMA

DaVinci PSP 03.01 release re-designed the EDMA driver (set of kernel APIs for use by other device drivers) by cleaning up unused structures and APIs and also renaming almost all APIs.

- davinci_request_dma changed to edma_alloc_channel. The new API does not require dev_name, channel and tcc to be passed. The allocated channel number is returned back to the caller. TCC and PaRAM allocation is handled by other APIs.
- davinci_set_dma_src_params changed to edma_set_src; davinci_set_dma_sest_params changed to edma_set_dest
- davinci_set_dma_src_index changed to edma_set_src_index; davinci_set_dma_dest_index changed to edma_set_dest_index
- davinci_set_dma_transfer_params changed to edma_set_transfer_params

- davinci_get_dma_params has been changed to edma_read_slot. davinci_set_dma_params has been changed to edma_write_slot. The argument should be "struct edmacc_param" (changed from struct paramentry_descriptor). TCC can be set using edma_write_slot and setting the TCC in the edma_param structure. Currently one-to-one mapping (of TCC to the channel) is allowed in the EDMA in the driver.
- davinci_start_dma changed to edma_start; davinci_stop_dma changed to edma_stop
- davinci_dma_link_lch changed to edma_link
- New APIs added to allocate/free PaRAM entries - edma_alloc_slot and edma_free_slot. If PaRAM allocation API is not called the driver assumes one-to-one mapping of PaRAM slots
- For davinci_free_dma, the new EDMA driver expects to call edma_clean_channel, edma_free_slot and then edma_free_channel

GPIO

GPIO driver has been rewritten to use the GPIO core infrastructure from the kernel. However there are no major changes to the APIs (back-end communication with the interrupts are handled with GPIO core).

- Need to call gpio_request() to start using a GPIO (this was not required earlier, we could directly call gpio_direction_xxx and gpio_write APIs)
- Provision to pass falling edge or rising edge information with the request_irq API. The IORESOURCE_IRQ_LOWEDGE and/or IORESOURCE_IRQ_HIGHEDGE can be passed for the "flags" argument.

Current driver (for DM355 only) has a limitation of using GPIO 0-9 direct/unbanked interrupts together with the banked interrupts. The default behavior of the driver is use banked interrupts for all GPIOs (including GPIO 0-9). In this case all GPIO interrupts work, provided they are available through pinmux. The DM355 SoC file - arch/arm/mach-davinci/dm355.c file can be configured to enable direct/unbanked interrupts for GPIO 0-9. However, with this change GPIO 10 - 103 in interrupt mode (through banked interrupts) would not work.

@@ -865,11 +865,12 @@ static struct davinci_soc_info davinci_soc_info_dm355 = {

```
        .intc_irq_prios         = dm355_default_priorities,
        .intc_irq_num           = DAVINCI_N_AINTC_IRQ,
        .timer_info             = &dm355_timer_info,
        .gpio_base              = IO_ADDRESS(DAVINCI_GPIO_BASE),
        .gpio_num               = 104,
-       .gpio_irq               = IRQ_DM355_GPIOBNK0,
+       .gpio_irq               = IRQ_DM355_GPIO0,
+       .gpio_unbanked          = 10,
        .serial_dev             = &dm355_serial_device,
        .sram_dma               = 0x00010000,
        .sram_len               = SZ_32K,
 };
```

Video drivers

The video display drivers and some other video related components in PSP 03.01 are an up-port of those present in LSP 2.10 [5]. For such components, the usage documentation provided with LSP 2.10 [5] will apply to PSP 3.01 as well. Here is a list of documents from older releases relevant to PSP 03.01. To access the documentation, download [6] and install the LSP collateral for the latest LSP 2.10 release.

- SPRUG90.pdf - LSP 2.00 DaVinci Linux AEW Driver
- SPRUG91.pdf - LSP 2.00 DaVinci Linux AF Driver
- SPRUG92.pdf - LSP 2.00 DaVinci Linux VPBE Frame Buffer Driver
- SPRUG95.pdf - LSP 2.00 DaVinci Linux Video Sysfs
- SPRUG96.pdf - LSP 2.00 DaVinci Linux V4L2 Display Driver

Video capture

The video capture driver supports capture from TVP5146 and MT9T031 sensor. The vpfe capture bridge driver (aka vpfe capture) is ported based on the v4l2 sub device framework. So all of the above mentioned decoder drivers are ported to this interface. The bridge device - sub device bus parameters such as interface type, hd, vd polarity etc are defined in the board specific file for respective evms (example, board-dm355-evm.c) and allows for easy customization. For vpfe capture driver, the platform data structure defines the vpfe configuration for the bridge driver as well sub devices and is used by the bridge driver to set the same in the bridge and sub devices (through platform_data).

The CCDC modules for a specific platform is registered with the vpfe capture. It registers a set of function calls that are called by vpfe capture driver to configure the ccdc module for capture.

The vpfe capture supports both MMAP and USERPTR IO. For high resolution frame buffer usage (beyond NTSC/PAL resolution) USERPTR (Both mmap-ed user address and virtual address pointers supported) IO is recommended since MMAP buffer allocation fails due to fragmentation.

For MT9T031 (only on DM355), the driver doesn't support VIDIOC_S_STD ioctl. So to capture a specific resolution frame, application needs to call VIDIOC_S_FMT ioctl. The vpfe capture from MT9T031 requires following bootargs to be set:- vpfe_capture.interface=1. By default, vpfe capture has this variable set to 0 for tvp5146 capture.

Phase 2 release adds support for TVP7002 driver (up-ported from LSP 2.xx releases) for HD capture on DM365. The VPFE capture driver on DM365 allows dynamic switching of TVP5146 and TVP7002 video input decoder sources. The HD catpure using MMAP application requires the following bootargs to be set:- vpfe_capture.bufsize=4147200. By default, vpfe capture has this variable set to (720*576*2) for SD resolution.

Note that when vpfe capture is used in the continuous (on the fly) mode and if second output from resizer is enabled, the location of the second image is at the end of first image (adjusted to kernel page boundary - 4096). In LSP 2.10 it was adjusted to 32 byte boundary.

All of the include files are now available under include/media/davinci directory. The vpfe capture header file to include is now called vpfe_capture.h (old name davinci_vpfe.h). VPFE_CMD_S_CCDC_PARAMS IOCTL is renamed to VPFE_CMD_S_CCDC_RAW_PARAMS. ccdc structure has some of the fields removed which was not really used. There are few fields renamed to meet open source coding style.

Video display

- Phase 1 Release: Only display driver supported is the old frame buffer display driver that was available in davinci git tree. This has been ported to DM355 and DM6446. It has only few features supported. It supports VID1 for displaying video and OSD0 and OSD1 for graphics. V4L2 display driver is currently not available.
- Phase 2 Release: The display driver architecture (encoder and display manager framework) from LSP 2.xx releases is up-ported to the GIT kernel. This architecture supports fbdev display driver for OSD and Video planes and supports V4L2 display driver for Video planes. It works similar to what is available in LSP2.10 release.

IPIPE - Previewer/Resizer

Phase 3 release adds support for IPIPE - Previewer/Resizer drivers, up-ported from LSP2.10. It works similar to what is available in LSP2.10.

H3A - AF/AEW

H3A drivers for DM355 and DM365 are ported from LSP2.10. It works similar to what is available in LSP 2.10.

Audio Driver

The amixer control for volume control across all outputs (master volume control) is managed through 'PCM Playback Volume' amixer control PSP 3.x

In LSP 2.x the master volume control is:

```
target$ amixer sget Master Volume Playback
Simple mixer control 'Master',0
  Capabilities: pvolume
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 127
  Mono:
  Front Left: Playback 126 [99%]
  Front Right: Playback 126 [99%]
```

In PSP 3.x the amixer master volume control is:

```
target$ amixer sget PCM Volume Playback
Simple mixer control 'PCM',0
  Capabilities: pvolume
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 127
  Mono:
  Front Left: Playback 80 [63%] [-23.50dB]
```

Front Right: Playback 80 [63%] [-23.50dB]

References

[1] http://www.arago-project.org

[2] http://www.codesourcery.com/sgpp/lite/arm/portal/release313

[3] http://www.codesourcery.com/sgpp/lite/arm/portal/release858

[4] http://www.mono-project.com/Downloads

[5] http://software-dl.ti.com/dsps/dsps_registered_sw/sdo_sb/targetcontent/psp/mv_lsp_2_10/index.html

[6] http://software-dl.ti.com/dsps/dsps_registered_sw/sdo_sb/targetcontent/psp/mv_lsp_2_10/02_10_00_14/exports/psp_02_10_00_14.bin

Article Sources and Contributors

DaVinci PSP 03.01 Linux Installation User Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=49496 *Contributors*: DanRinkes, M-karicheri2, Nsnehaprabha, SekharNori, Sudhakar.raj

Image Sources, Licenses and Contributors

Image:TIBanner.png *Source*: http://processors.wiki.ti.com/index.php?title=File:TIBanner.png *License*: unknown *Contributors*: Nsnehaprabha

License

License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

1. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
2. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
3. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
4. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
5. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
6. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.