U-boot Porting Development Guide

Version: 1.2.1.1

Release date: 2022-06-01

Contents

Revision History

Revision	Date	Description
0.0.0.1	2022/06/01	Initial

1 Disclaimer

Terms and Conditions

The document and all information contained herein remain the CVITEK Co., Ltd' s ("CVITEK") confidential information, and should not disclose to any third party or use it in any way without CVITEK' s prior written consent. User shall be liable for any damage and loss caused by unauthority use and disclosure.

CVITEK reserves the right to make changes to information contained in this document at any time and without notice.

All information contained herein is provided in "AS IS" basis, without warranties of any kind, expressed or implied, including without limitation mercantability, non-infringement and fitness for a particular purpose. In no event shall CVITEK be liable for any third party' s software provided herein, User shall only seek remedy against such third party. CVITEK especially claims that CVITEK shall have no liable for CVITEK' s work result based on Customer' s specification or published shandard.

Contact Us

Address Building 1, Yard 9, FengHao East Road, Haidian District, Beijing, 100094, China

Building T10, UpperCoast Park, Huizhanwan, Zhancheng Community, Fuhai Street, Baoan District, Shenzhen, 518100, China

Phone +86-10-57590723 +86-10-57590724

Website https://www.sophgo.com/

Forum https://developer.sophgo.com/forum/index.html

2 Function Overview

2.1 Purpose

cv180x/cv181x uses U-boot-2021.10 as Bootloader on the its EVB. When different peripheral processors are configured (i.e.the development version and the public version are different), you need to modify the U-boot related codes, mainly including registers, configuration files and drivers.

2.2 U-boot Directory Structure

The following table lists the common modification directories and files:

Directory	Description
arch	RISC-V processor development related code
arch/arm/dts or arch/riscv/dts	Linux/uboot shared DTS configuration file. The actual DTS file path (not inside the u-boot folder) is sdk _source/build/boards/$(CHIP_ARCH)/$(BOARD)/dts_riscv (riscv)/$(BOARD).dts (for cv180x) ex. cv180x/cv181x processor series EVB board named cv1800b_wevb_0008a_spinor sdk_source/build/boards/cv180x/cv1800b_wevb_0008a_spinor/ dts_riscv/$(BOARD).dts
configs	uboot config configuration file. The actual file path for configs is sdk_source/build/boards/$(CHIP_ARCH)/$(BOARD)/u-boot/ cvitek_$(BOARD)_defconfig defconfig: uboot native or new configuration ex. cv180x/cv181x processor series EVB board named cv1800b_wevb_0008a_spinor sdk_source/build/boards/cv180x/cv1800b_wevb_0008a_spinor/ u-boot/cvitek_cv1800b_wevb_0008a_spinor_defconfig
Board	Relavent codes of various SOC processor manufacturers, board-side settings that need to be configured after the EVB is powered on. cvitek.h : set GPIO definition and different EVB differences cvi_board_init.c: control EVB board segment I/O, PINMUX peripheral processor settings The actual file path for board.c/cvitek.h is sdk_source/build/boards/$(CHIP_ARCH)/$(BOARD)/u-boot/ cvi_board_init.c sdk_source/build/boards/$(CHIP_ARCH)/$(BOARD)/u-boot/ cvitek.h ex. cv180x/cv181x processor series EVB board named cv1800b_wevb_0008a_spinor sdk_source/build/boards/cv180x/cv1800b_wevb_0008a_spinor/ u-boot/cvi_board_init.c sdk_source/build/boards/cv180x/cv1800b_wevb_0008a_spinor/ u-boot/cvitek.h
Include	Header files
Include/configs	cv180x-asic.h/cv181x-asic.h set boot command/configuration.
cmd	Uboot console command function codes
drivers	Etherent, usb, storage and other related drivers.

3 U-boot Transplant

3.1 U-boot Hardware Environment

The peripheral processors on cv180x/cv181x EVB include DDR, eMMC, SPI NAND Flash and SPI NOR Flash. Please refer to *CV181x/CV180xB/C Hardware Design User Guide V1.0* for all models.

3.2 Pin Configuration (Pinmux)

For different EVBs and different peripherals, the initialization settings can be done in cvi_board_init.c.

```
$ cat build/boards/cv180x/cv1800b_wevb_0008a_spinor/u-boot/cvi_board_init.c
int cvi_board_init(void)
{
      PINMUX_CONFIG(PAD_MIPIRX1P, IIC1_SDA);
      PINMUX_CONFIG(PAD_MIPIRX0N, IIC1_SCL);
      PINMUX_CONFIG(PAD_MIPIRX1N, XGPIOC_8);
      PINMUX_CONFIG(PAD_MIPIRX0P, CAM_MCLK0);
      return 0;
}
```

3.3 Compile U-boot

The operation of compiling U-boot is as follows:

- Read compilation environment variables (take cv1800b_wevb_0008a_spinor as an example)

```
$ source build/cvisetup.sh
  ------------------------------------------------------------------------
    Usage:
    (1) menuconfig - Use menu to configure your board.
        ex: $ menuconfig
```

(continues on next page)

(continued from previous page)

```
    (2) defconfig $CHIP_ARCH - List EVB boards($BOARD) by CHIP_ARCH.
       ** cv183x ** -> ['cv1829', 'cv1832', 'cv1835', 'cv1838', 'cv9520',
↪'cv7581']
       ** cv182x ** -> ['cv1820', 'cv1821', 'cv1822', 'cv1823', 'cv1825',
↪'cv1826', 'cv7327', 'cv7357']
       **   cv181x ** -> ['cv181x', 'cv1823a', 'cv1821a', 'cv1820a', 'cv1811h',
↪'cv1811c', 'cv1810c', 'cv1812h']
       ** cv180x ** -> ['cv180x', 'cv1800b', 'cv1800c', 'cv1801b', 'cv1801c']
        ex: $ defconfig cv183x

    (3) defconfig $BOARD - Choose EVB board settings.
        ex: $ defconfig cv1835_wevb_0002a
        ex: $ defconfig cv1826_wevb_0005a_spinand
        ex: $ defconfig cv181x_fpga_c906
  ------------------------------------------------------------------------
```

- Select EVB cv1800b_wevb_0008a_spinor

```
$ defconfig cv1800b_wevb_0008a_spinor
Run defconfig function
Loaded configuration '/workspace/build/boards/cv180x/cv1800b_wevb_0008a_spinor/
↪cv1800b_wevb_0008a_spinor_defconfig'
No change to configuration in '.config'
Loaded configuration '.config'
====== Environment Variables =======
  PROJECT: cv1800b_wevb_0008a_spinor, DDR_CFG=ddr2_1333_x16
  CHIP_ARCH: cv180x, DEBUG=0
  SDK VERSION: musl_riscv64, RPC=0
  ATF options: ATF_KEY_SEL=default, BL32=1
  Linux source folder: linux_5.10, Uboot source folder: u-boot-2021.10
  CROSS_COMPILE_PREFIX: riscv64-unknown-linux-musl-
  ENABLE_BOOTLOGO: 0
  Flash layout xml: /workspace/build/boards/cv180x/ cv1800b_wevb_0008a_spinor/
↪partition/partition_spinor.xml
  Sensor tuning bin: gcore_gc4653
  Output path: /workspace/master/install/ soc_cv1800b_wevb_0008a_spinor
```

- Compile U-boot

```
$ build_uboot
[TARGET] u-boot-dts
......
[TARGET] u-boot-build
......
```

- Get fip_spl.bin adn fip.bin (with bootloader+uboot inside)

```
$ ls install/soc_cv1800b_wevb_0008a_spinor/fip.bin
```

(continues on next page)

(continued from previous page)

```
install/soc_cv1800b_wevb_0008a_spinor/fip.bin
$ ls install/soc_cv1800b_wevb_0008a_spinor/fip_spl.bin
install/soc_cv1800b_wevb_0008a_spinor/fip_spl.bin
```

Since the native u-boot compiled u-boot.bin cannot be burned directly into FLASH, we adopt the Firmware Image Package (FIP) method in ARM Trusted Firmware Design to encapsulate uboot.bin in fip.bin. fip_spl.bin contains several images that need to be loaded during the fast boot process.

4 U-boot Burn Update

4.1 Overview

U-boot burn update will need to burn the entire fip.bin (bootloader + uboot), while the bootloader comes with different DDR initialization parameters, which can be configured via SDK menuconfig when EVB is selected.

`Please refer to the SDK_Compilation_and_Usage_Guide - 1.4.2.2.docx`

4.2 U-boot Through Bootrom

`Please refer to Cvitek Bare and Non-Bare Processor Burning Upgrade Operation Guide_v1.2.1.docx`

4.3 Flash U-boot Burn Update

4.3.1 SPI NOR Flash Burn Update

`Please refer to Cvitek Bare and Non-Bare Processor Burning Upgrade Operation Guide_v1.2.1.docx`

4.3.2 SPI NAND Flash Burn Update

`Please refer to Cvitek Bare and Non-Bare Processor Burning Upgrade Operation Guide_v1.2.1.docx`

4.3.3 eMMC Burn Update

Please refer to Cvitek Bare and Non-Bare Processor Burning Upgrade Operation Guide_v1.2.1.docx