



Linux Development Environment User Guide

Version: 0.0.0.1

Release date: 2022-06-01

Copyright © 2020 CVITEK Co., Ltd. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of CVITEK Co., Ltd.

Contents

1	Disclaimer	2
2	Development environment	3
2.1	Purpose	3
2.2	How to compile the kernel	3
3	U-boot setup	5
4	Linux Kernel	6
4.1	Set Kernel DTS	6
4.2	Setup kernel configuration	7
5	Root file system (rootfs)	9
5.1	Introduction about root file system	9
5.2	Busybox support	9

Revision History

Revision	Date	Description
0.0.0.1	2022/06/01	draft

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 Development environment

2.1 Purpose

This document describes the Linux development environment. Setting up of Linux development environment, U-boot, Linux kernel, root file system (rootfs) , programming of kernel and root file system burning, as well as creating a network development environment and starting Linux development. This document provides a user terminal that can quickly build a Linux environment and port self-developed applications to Linux above the operating system.

2.2 How to compile the kernel

- To compile the SDK in the ubuntu environment on the HOST side, you need to install the following tools:

Please refer to the SDK compilation and use instructions_V1.0.docx to build compilation environment

- 1) Set environment variables (example: v1800b_wevb_0008a_spinor)

```

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

```

(continues on next page)

(continued from previous page)

```

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

```

- 2) 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

```

- 3) Compile linux kernel

```

$ build_kernel
[TARGET] kernel-dts
.....
[TARGET] kernel-build
.....

```

- 4) Generate images boot.{spinor, spinand, emmc}

```

$ ls install/soc_cv1800b_wevb_0008a_spinor/boot.spinor
install/soc_cv1800b_wevb_0008a_spinor/boot.spinor

```

3 U-boot setup

Please refer to the U-boot porting application development `guide_v1.2.0.1.docx`

4 Linux Kernel

The kernel code can be found in the In sdk_source directory

```
sdk_source/linux // version 4.19, cv182xA, ca53 32bit Processor
sdk_source/linux_5.10 // cv180x,cv181x C906B 64 bit Processor
```

4.1 Set Kernel DTS

If you want to increase or decrease the kernel module, you can do it by modifying DTS(*1). Each EVB will have a dts file to define its device tree. Such as cv1800b_wevb_0008a_spinor, its DTS file is defined in the file path as follows:

```
$ cat build/boards/cv180x/cv1800b_wevb_0008a_spinor/dts_riscv/
cv1800b_wevb_0008a_spinor.dts
/dts-v1/;
#include "cv180x_base_arm.dtsi"
#include "cv180x_asic_bga.dtsi"
#include "cv180x_asic_spinor.dtsi"
#include "cv180x_default_memmap.dtsi"

/ {
// add your customized device description
};
```

The above *.dtsi (device tree source include files) is the preset value of the processor, and it is not recommended to change it directly. To modify the preset value, it is recommended to use the(delete node)method.

(*1) U-boot and kernel use common DTS

4.2 Setup kernel configuration

If you want to modify the configuration of the kernel, you can directly modify the kernel configuration file. Such as cv1800b_wevb_0008a_spinor, The defconfig file is defined in the file path as follows

```
$ cat build/boards/cv180x/cv1800b_wevb_0008a_spinor/linux/cvitek_cv1800b_wevb_0008a_spinor_defconfig
# CONFIG_SWAP is not set
CONFIG_SYSVIPC=y
CONFIG_POSIX_MQUEUE=y
CONFIG_NO_HZ_IDLE=y
CONFIG_HIGH_RES_TIMERS=y
CONFIG_PREEMPT=y
CONFIG_IKCONFIG=y
CONFIG_IKCONFIG_PROC=y
CONFIG_LOG_BUF_SHIFT=15
CONFIG_CC_OPTIMIZE_FOR_SIZE=y
```

- Example of using the method to modify the defconfig file (add support for SPI driver)

```
#
# SPI drivers
#
# CONFIG_SPI is not set
# CONFIG_SPI_MASTER is not set
# CONFIG_SPI_DESIGNWARE is not set
# CONFIG_SPI_DW_MMIO is not set
# CONFIG_SPI_SPIDEV is not set
CONFIG_SPI=y
CONFIG_SPI_MASTER=y
CONFIG_SPI_DESIGNWARE=y
```

- Use the way like command line—setconfig_kernel

```
$ setconfig_kernel SPI=y
$ setconfig_kernel SPI_MASTER=y
$ setconfig_kernel SPI_DESIGNWARE=y
```

- Use the way like Graphic user interface line - menuconfig_kernel

```
$ menuconfig kernel
.config - Linux/riscv 5.10.4 Kernel Configuration
Linux/riscv 5.10.4 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->
(or empty submenus ----). Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] built-in [ ] excluded <M> module < >
module capable
  General setup --->
    [*] MMU-based Paged Memory Management Support (NEW)
    (10) Maximum zone order
        SoC selection --->
        Platform type --->
        Kernel features --->
        Boot options --->
        Power management options --->
        Firmware Drivers --->
    [*] Cvitek SoC Family
<Select>  < Exit >  < Help >  < Save >  < Load >
```

5 Root file system (rootfs)

5.1 Introduction about root file system

Please refer to the `SDK_Compilation_and_Usage_Guide - Chapter 5 root file system (rootfs)`

5.2 Busybox support

At present, the file system uses BusyBox v1.27.1. If you need to update busybox, you can put the compiled busybox in the following path:

```
$ ramdisk/rootfs/common_musl_riscv64/bin/busybox // CV181X
$ ramdisk/rootfs/common_musl_riscv64/bin/busybox // CV180X
```