

# **N9H2x/N329xx Series MCU Linux 2.6.35 BSP User's Manual V1.6**

***Publication Release Date: Feb. 2020***

---

**Support Chips:**  
N9H2x/N329XX Series

**Support Platforms:**  
Linux

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from the Nuvoton Technology Corp.

Nuvoton Technology Corp. All rights reserved.

# Table of Contents

1.	GENERAL DESCRIPTION .....	6
2.	DEVELOPMENT ENVIRONMENT SETUP .....	7
2.1.	N9H2x//N329xx Series DEV Board Jumper Setting .....	7
3.	CHANGE IN THIS BSP .....	8
4.	BSP INSTALLATION .....	9
4.1.	Before Installation .....	9
	CentOS .....	9
	Ubuntu .....	10
4.2.	Download BSP Relative Files .....	10
4.3.	Install ARM Linux GNU Tool Chain .....	10
4.4.	Install N9H2x/N329xx Series Linux BSP .....	11
5.	KERNEL COMPILE AND DOWNLOAD TO BOARD ..	13
5.1.	Load Default Configuration .....	13
	N9H20K3/N32903 .....	14
	N9H20K5 /N32905 .....	14
	N32915 .....	14
	N32916 .....	14
	N32925 .....	14
	N9H26/N32926 .....	14
5.2.	Kernel Compilation .....	15
5.3.	Change Booting Device .....	15
5.4.	Download Kernel to Board .....	16
6.	KERNEL CONFIGURATION .....	17
6.1.	System configuration .....	17
	Code maturity level selection .....	17
	Configure initramfs .....	17
	Loadable module support .....	17
	System Type .....	17
	Boot options .....	18
6.2.	Device driver configuration .....	18
	UART support .....	18
	SD/SDIO card support (SDIO driver is for N9H26/N3292x only) .....	18

NAND Flash support.....	19
SPI device support .....	19
SPI flash support (using MTD support) .....	19
Video/OSD/PWM support.....	20
Audio playback/record support.....	20
AAC support (for N9H26/N3292x only) .....	21
Video Capture support.....	21
JPEG codec support .....	22
H264 decoder support (for N3291x only) .....	22
H264 codec support (for N9H26/N3292x only) .....	22
VPE support (for N3291x and N9H26/N3292x only) .....	23
BLT support (for N9H20/N3290x and N9H26/N3292x only) .....	23
OVG support (for N3291x only) .....	23
Keypad (GPIO) support (for N9H20/N3290x and N3291x only) .....	23
Keypad (KPI) support (for N3291x only).....	23
Touch Screen/Voltage detection support (for N9H20/N3290x and N3291x only) .....	24
Keypad/Touch Screen/Voltage detection support (for N9H26/N3292x only) .....	24
USB host support (EHCI driver is for N9H26/N3292x only).....	24
USB device support.....	25
Watchdog timer support .....	25
RTC support.....	25
I2C support .....	25
GPIO support.....	26
AES support (for N3291x and N9H20/N3292x only) .....	26
EMAC support (for N9H26/N3292x only).....	26
<b>6.3. File system support.....</b>	<b>27</b>
FAT-based file systems(MS-DOS, VFAT) support.....	27
ROM file system support .....	27
JFFS2 file system support .....	27
/proc and sysfs file system support .....	27
<b>6.4. Network support.....</b>	<b>28</b>
TCP/IP support.....	28
Wireless support .....	28
<b>7. HARDWARE AND DRIVER SOURCE LOCATION.....</b>	<b>29</b>
<b>8. SAMPLE APPLICATIONS.....</b>	<b>31</b>
8.1. Cross Compilation Makefile .....	33
<b>9. ADVANCE CONTROL IN LINUX BSP .....</b>	<b>35</b>
9.1. How to enable core dump feature? .....	35
9.2. How to enter to suspend mode and resume? .....	35

N9H2X/N3290x .....	35
N3291x .....	35
N9H26/N3292x.....	35
9.3. How to dynamic change CPU and PLL clocks? .....	36
N9H20/N3290x.....	36
N3291x .....	36
N9H26/N3292x.....	36
10. REVISION HISTORY .....	37

# 1. General Description

This is a Linux BSP for Nuvoton N9H2x/N329xx series MCU. Nuvoton N9H2x/N329xx series MCU are built around an outstanding CPU core: the 32-Bit ARM926EJS RISC processor designed by Advanced RISC Machines, Ltd. The ARM926EJS core, offers 8/16K-byte I-cache and 8/16K-byte D-cache with MMU, is a low power, general-purpose integrated circuits. This series of micro-controllers are suitable for a high end, high performance and low cost related products as well as general purpose applications.

This Linux BSP contains following items:

- Linux 2.6.35.4 kernel source code including the drivers for N9H2x/N329xx series MCU.
- Source code of sample applications for different interfaces, busybox, tslib, and other open source applications
- ARM Linux GNU tool chain
- Turbo Writer tool
- Relative documents

## 2. Development environment setup

This BSP only provides cross development tool chain in Linux environment. So Linux platform is a must to build Linux kernel/applications using the cross compiling tool chain in BSP. This Linux platform could be a dedicate Linux PC or running on virtual machine.

Windows PC can communicate with N9H2x/N329xx series DEV board via different communication interfaces, such as USB. As well as debug port, JTAG. Above interfaces could be used to load binary file to DEV board for execution. JTAG interface is used for chip level debug. USB interface is used by Turbo Writer to write SD/NAND/SPI flash. UART interface is used to get the running message.

Figure 2-1 Development Environment is an example of development environment.

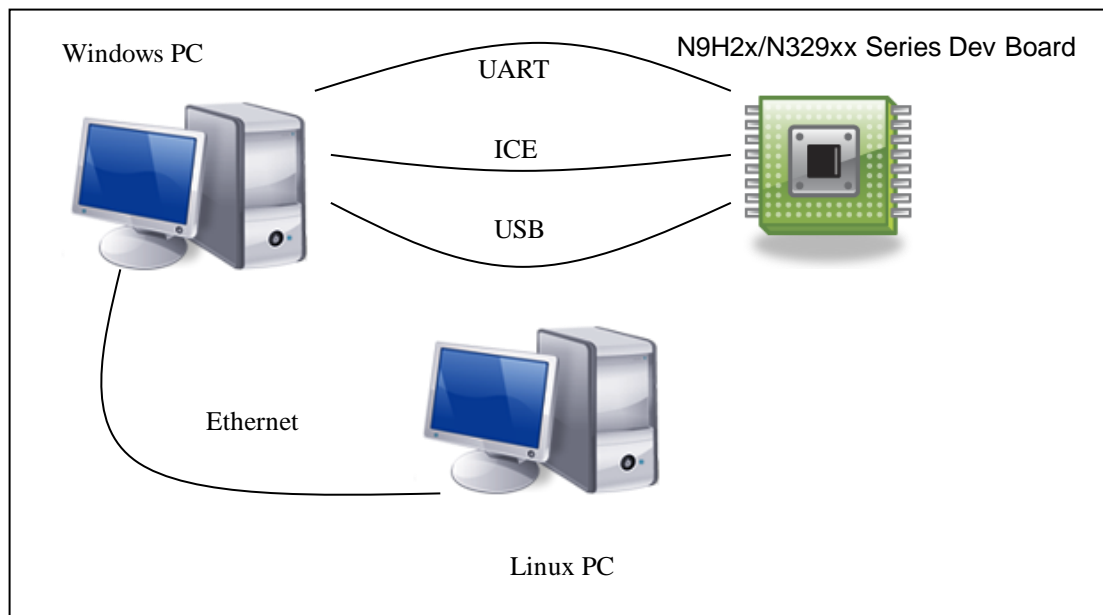


Figure 2-1 Development Environment

### 2.1. N9H2x//N329xx Series DEV Board Jumper Setting

Due to the pin number limitation, some interfaces in N9H2x/N329xx series MCU share the same group of pins. The setting of Linux kernel and DEV board must be consistent to make these interfaces work correctly. Chapter 4 will describe how to configure the interfaces in Linux kernel. Please refer to the *DEV Board User's Manual* for the correct switch/jumper settings on DEV board.

### 3. Change in this BSP

For users who has used previous N9H2x/N329xx series Linux BSP based on Linux 2.6.17.14, below list the major changes:

- Linux kernel upgrade to 2.6.35.4
- Provides ALSA driver, OSS driver is no longer supported
- SD driver moved from /drivers/scsi to /drivers/mmc
- SPI flash no longer control by user application as char device. Now registered as MTD device
- Removed mass storage application, use USB gadget mass storage kernel module instead
- Removed hotplug application, use mdev applet in busybox
- Support SDIO WiFi devices
- Support N9H2x/N329xx series which with only 8MB bytes memory
- Root FS folder is renamed to rootfs-2.6.35



## 4. BSP Installation

The BSP contains four directories. The contents of each directories describes in following table:

Directory Name	Contents
ARM_toolchain	Tar ball of ARM cross tool chain and install script
BSP	Tar ball of Linux related stuffs. Including kernel source code, sample applications, sample rootfs.
Documents	BSP related documents
../Utility	Windows tool

### 4.1. Before Installation

The development environment of N9H2x/N329xx series BSP is based on Linux PC. Nuvoton has tested Linux BSP in CentOS and Ubuntu Linux distributions on x86 PCs. And it should also work well on other Linux distributions.

#### CentOS

##### Root Privilege

CentOS is recommend due to you can change to be root account or use root account to login system directly. **In Linux BSP development, root privilege is necessary in some procedures.** Such as execute the installation script or extract the RootFS file to create a device node.

##### Missing Packages in CentOS 8.1 64bit

CentOS does not install some packages by default and which are mandatory for N9H2x/N329xx series Linux BSP. Below table listed missing packages which are mandatory during development.

Package name	Function
patch	Utility for modifying/upgrading files
make	A GNU tool which simplifies the build process for users
gcc	Various compilers (C, C++, Objective-C, ...)
ncurses-devel	Development files for the ncurses library
glibc.i686	The GNU libc libraries (for 32bit cross compiler)
libstdc++-8.3.1-4.5.el8.i686	GNU Standard C++ Library (for 32bit cross compiler)

CentOS users could use **`sudo yum install -y <package name>`** command to execute these packages.

## Ubuntu

### Root Privilege

By default, Ubuntu does not set the root password. So it is impossible to use **`su`** command to switch to root and do some jobs require root privilege before set the root password. Users of Ubuntu could use **`sudo`** command to execute a single command with root privilege instead.

Please use following command: **`sudo <user> admin`** to grant the super user privilege for **`<user>`**.

### Missing Packages in Ubuntu 18.04 LTS 64bit

Ubuntu does not install some packages by default and which are mandatory for N9H2x/N329xx series Linux BSP. Below table listed missing packages which are mandatory during development.

Package name	Function
make	Utility for directing compilation
gcc	GNU C compiler
libncurses5-dev	Developer's libraries for ncurses
libc6-dev-i386	GNU C Library: 32-bit development libraries for AMD64 (for 32bit cross compiler)

Ubuntu users could use **`sudo apt-get install <package name>`** command to execute these packages. If GUI interface is preferred, use Synaptic Package Manager to install the missing packages.

---

## 4.2. Download BSP Relative Files

To do N9H2x/N329xx series BSP installation, you should need to download ARM\_toolchain and latest released version in BSP folder to your Linux PC. You could find these files in Nuvoton's FTP server.

---

## 4.3. Install ARM Linux GNU Tool Chain

If your Linux PC has not installed the arm\_linux tool chain, please install the tool chain first. The produce is very simple, you just need to run the installation script (eg: `./arm_linux_4.8_install.sh`), it will install the tool chain into `/usr/local/arm_linux_4.8`.

If your Linux PC has already installed the arm\_linux\_4.8 tools, the installation script will ask whether or not to remove the existing tool chain. You can type "yes" if you want to update the tool chain.



```
# sudo ./w55fa93bsp_install.sh
```

```

root@localhost:/home/bsp/W55FA93BSP2.6.35_130611
File Edit View Search Terminal Help
[root@localhost W55FA93BSP2.6.35_130611]# ./w55fa93bsp_install.sh
Please enter absolute path for installing BSP (eg: /home/w55fa93bsp-2.6.35)
Press Enter key directly will use the default path "/home/w55fa93bsp-2.6.35"

Now installing w55fa93 Linux BSP to "/home/w55fa93bsp-2.6.35"
The folder "/home/w55fa93bsp-2.6.35" is not exist, create it now.
please wait for a while, it will take some time
Extracting linux-2.6.35.4.tar.gz
Extracting patch-2.6.35.tar.gz
Extracting image.tar.gz
Extracting rootfs-2.6.35.tar.gz
Extracting rootfs-2.6.35_N32903.tar.gz
Extracting applications.tar.gz
Now patching the kernel linux-2.6.35.4
are you sure to do patch 0-2?[y/n] patching file /home/w55fa93bsp-2.6.35/patch-2
.6.35/w55fa93-kernel-2.6.35-000.patch
patching file /home/w55fa93bsp-2.6.35/patch-2.6.35/w55fa93-kernel-2.6.35-001.pat
ch
patching file /home/w55fa93bsp-2.6.35/patch-2.6.35/w55fa93-kernel-2.6.35-002.pat
ch
do patches successfully
Now changing the directory and file attributes for user "root"
Installing w55fa93 Linux BSP successfully
[root@localhost W55FA93BSP2.6.35_130611]#

```

Fig 2 Linux BSP installation

This installation script will ask for the **absolute path** for install kernel and applications. The table below listed the item will be installed in the specified location. If you do not specify the install path, the default path should be the "/home/w55fa9xbsp2.6.35".

Directory Name	Description
applications	Source code of sample applications, busybox, tslib.
Image	Pre-build kernel image and rootfs
linux-2.6.35.4	Kernel source code
rootfs-2.6.35	Root file system
rootfs-2.6.35_N32903	Root file system for N9H20K3/N32903, it is only provided in N9H20Kx/N3290x BSP
rootfs-2.6.35_N32903_WiFi	Root file system for N32903 supports SDIO WiFi, it is only provided in N9H20Kx/N3290x BSP
patch-2.6.35	Kernel patches

## 5. Kernel Compilation and Download to Board

---

### 5.1. Load Default Configuration

Nuvoton provides default kernel configuration for each MCU. It is strongly urged that users load the default setting according to the target MCU with board number before first time to build the kernel. To do so, use “make <mcu\_name>\_defconfig” in linux-2.6.35.4/ directory to load default kernel configuration. Below are the examples for demo boards.

### **N9H20K3/N32903**

*NHS-W55FA93-1-IN-1M1x demo board: make n32903\_defconfig*

*NHS-W55FA93-1-IN-1M1x demo board supports SDIO WiFi: make n32903\_wifi\_defconfig*

*HMI-N9HK3 demo board: make n9h20k3\_defconfig*

### **N9H20K5 /N32905**

*NHS-W55FA93-1-IN-1M1x demo board: make N32905\_defconfig*

*HMI-N9H20K5 demo board: make n9h20k5\_defconfig*

### **N32915**

*NHS-W55FA95-1-YS-1M56 demo board: make n32915\_defconfig*

*NHS-W55FA95-1-YS-1M55 demo board: make n32915\_defconfig*

### **N32916**

*NHS-W55FA95-1-YS-1M56 demo board: make w55fa95\_defconfig*

*NHS-W55FA95-1-YS-1M55 demo board: make w55fa95\_defconfig*

*NHS-W55FA95-1-YS-1M52 demo board: make w55fa95\_YS1M52\_defconfig*

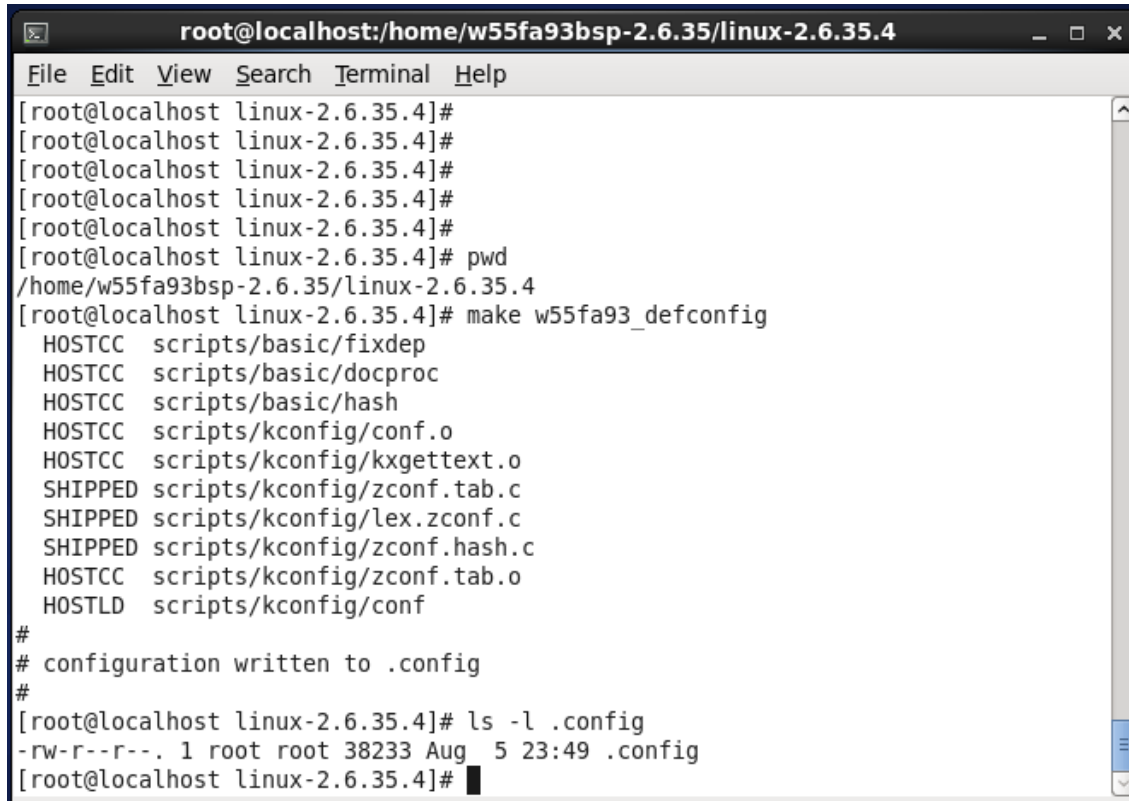
### **N32925**

*NHS-N3292x-1-PC-2D0x demo board: make n32925\_defconfig*

### **N9H26/N32926**

*NHS-N3292x-1-PC-2D0x demo board: make w55fa92\_defconfig*

*HMI-N9H26 demo board: make n9h26\_defconfig*



```

root@localhost:/home/w55fa93bsp-2.6.35/linux-2.6.35.4
File Edit View Search Terminal Help
[root@localhost linux-2.6.35.4]#
[root@localhost linux-2.6.35.4]#
[root@localhost linux-2.6.35.4]#
[root@localhost linux-2.6.35.4]#
[root@localhost linux-2.6.35.4]#
[root@localhost linux-2.6.35.4]# pwd
/home/w55fa93bsp-2.6.35/linux-2.6.35.4
[root@localhost linux-2.6.35.4]# make w55fa93_defconfig
HOSTCC  scripts/basic/fixdep
HOSTCC  scripts/basic/docproc
HOSTCC  scripts/basic/hash
HOSTCC  scripts/kconfig/conf.o
HOSTCC  scripts/kconfig/kxgettext.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/lex.zconf.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC  scripts/kconfig/zconf.tab.o
HOSTLD  scripts/kconfig/conf
#
# configuration written to .config
#
[root@localhost linux-2.6.35.4]# ls -l .config
-rw-r--r--. 1 root root 38233 Aug  5 23:49 .config
[root@localhost linux-2.6.35.4]#

```

Fig 3 Load default configuration

## 5.2. Kernel Compilation

After load the default configuration, type “make” to compile the kernel. The newly created kernel will be copied into ../image/ directory. If the rootfs uses ROMFS type, the compressed file of rootfs will be also copied to the same folder.

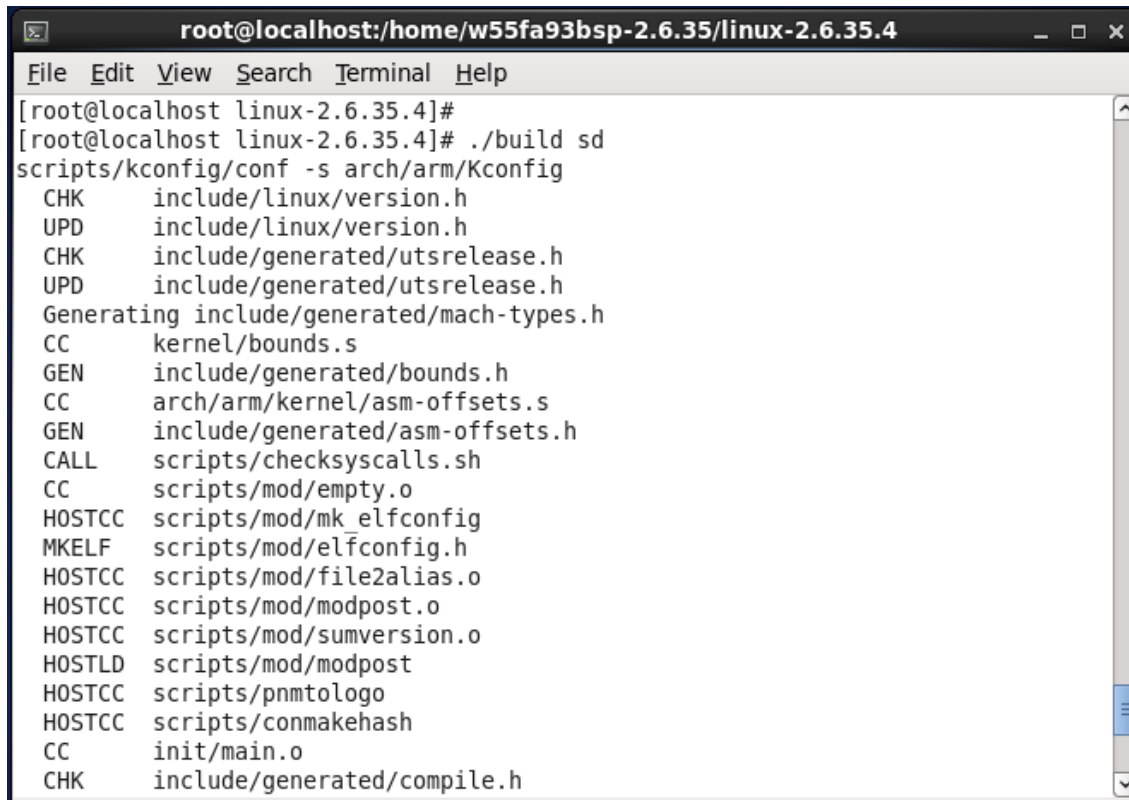
**If you set the NAND driver or USB Gadget driver to be a kernel module, please type “./build” to compile the kernel when you modify relative drivers.** It will also update the kernel modules in rootfs, or you will still use previous kernel module after kernel booting.

**Please notice if there is no any on board NAND chip, you need to disable “on board NAND” option in kernel configuration.** Enable this option may cause a GNAND initial failed issue if there is no any on board NAND chip. For how to change the setting, please refer to **NAND Flash support** in section 6.1.

## 5.3. Change Booting Device

N9H2x/N329 series BSP supports SD/NAND/SPI booting devices. You can type “./build [sd/nand/spi]” to build kernel with specified booting device. Once you have selected a booting

device, if you do not want to change it, you can just type "make" or "./build" command to build kernel in next time.



```

root@localhost:/home/w55fa93bsp-2.6.35/linux-2.6.35.4
File Edit View Search Terminal Help
[root@localhost linux-2.6.35.4]#
[root@localhost linux-2.6.35.4]# ./build sd
scripts/kconfig/conf -s arch/arm/Kconfig
CHK      include/linux/version.h
UPD      include/linux/version.h
CHK      include/generated/utsrelease.h
UPD      include/generated/utsrelease.h
Generating include/generated/mach-types.h
CC      kernel/bounds.s
GEN      include/generated/bounds.h
CC      arch/arm/kernel/asm-offsets.s
GEN      include/generated/asm-offsets.h
CALL     scripts/checksyscalls.sh
CC      scripts/mod/empty.o
HOSTCC   scripts/mod/mk_elfconfig
MKELF    scripts/mod/elfconfig.h
HOSTCC   scripts/mod/file2alias.o
HOSTCC   scripts/mod/modpost.o
HOSTCC   scripts/mod/sumversion.o
HOSTLD   scripts/mod/modpost
HOSTCC   scripts/pnmtologo
HOSTCC   scripts/conmakehash
CC      init/main.o
CHK      include/generated/compile.h
  
```

*Fig 4 Use script to change booting device*

## 5.4. Download Kernel to Board

If boot loader execute NVT Loader first and run to kernel, the kernel image should be put into SD1-1 or NAND1-1 FAT partition. Or the kernel image need be burned into SD/NAND/SPI system area via Turbo Writer tool. To download image to SDRAM, SD card, NAND flash, or SPI flash, please refer to Turbo Writer User's Manual.pdf.



## 6. Kernel Configuration

This chapter describes how to configure system, drivers, file system, network... etc. Please type “make menuconfig” command in linux-2.6.35.4/ directory to enter kernel configuration menu.

### 6.1. System configuration

#### **Code maturity level selection**

General setup ->

[\*] Prompt for development and/or incomplete code/drivers

#### **Configure initramfs**

Initramfs is used as the root file system. This string should be the directory name for building the initramfs image.

General setup ->

(../rootfs-2.6.35) Initramfs source file(s)

#### **Loadable module support**

Some kernel modules could be installed into kernel after system boot up.

Loadable module support ->

--- Enable loadable module support

[\*] Module unloading

#### **System Type**

In the System Type sub menu, users need to configure the correct clock rate and kernel load address.

System Type ->

ARM system type (Nuvoton N9H2x/N329xx/W55FA93 MCU) ->

N9H2x/N329xx/W55FA93 Machines ->

[\*] N9H2x/N329xx/W55FA93 DEV-Nuvoton Evaluation Kit

External Crystal (12MHz) ->

[\*] CPU clock set by boot code

Kernel Load Address (0x00000000) ->

### **Boot options**

Boot options allow system features, such as console attributes, system ram size, to be specified during compilation. Default configuration will set rdinit to /sbin/init, ttyS0 running at 115200 8-N-1 as console port, and 32MB system memory. So default boot option looks like:

Boot options ->

```
(root=/dev/ram0 console=ttyS1,115200n8 rdinit=/sbin/init mem=32M vt.global_cursor_default=0)
```

## **6.2. Device driver configuration**

This section describes how to enable driver for each interfaces.

### **UART support**

By default, HUART option is disabled due to it conflicts with the JTAG interface.

Device Drivers ->

Character devices ->

Serial drivers ->

```
<*> Nuvoton N9H2x/N329xx/W55FA93 Serial port support
[ ]   Nuvoton N9H2x/N329xx/W55FA93 HUART support
[*]   Support for console on N9H20x/N3290x/W55FA93 serial port
```

### **SD/SDIO card support (SDIO driver is for N9H26/N3292x only)**

SD and SDIO are two lps using the same engine. So SD driver also could be used to connect a SDIO WiFi device. Users can adjust the value of SD Disk Mount Delay to control the SD response time out value. The longer delay makes SD disk mount slower but more stable.

Device Drivers ->

```
<*> MMC/SD/SDIO card support ->
--- MMC/SD/SDIO card support
<*>   MMC block device driver
[*]    Use bounce buffer for simple hosts
<*>   Nuvoton N9H2x/N329xx/W55FA92 SIC/SD Card support
      SD port select (support SD0 port only) ->
[ ]    SD0 Write protect (GPA0) support
<*>   Nuvoton N9H2x/N329xx/W55FA92 SDIO Card support
      SDIO port select (support SDIO1 port only) ->
      SDIO1 Card Detect Pin select (GPG4 (FA92 default)) ->
      SDHC clock select (Use 24MHz clock for SDHC card) ->
(0x2FFF) SD Disk Mount Delay
```

### **NAND Flash support**

NAND driver source code is included in BSP. However, physical to logical mapping scheme and wear leveling algorithm (gnand.ko) is not GPL licensed, and only released as a kernel module. So NAND driver also needs to be build as a kernel module. You can find the NAND driver kernel module in drivers/scsi/ uvoton\_nand/nanddrv.ko. This driver supports both SLC and MLC NAND flash. To use NAND flash with FAT file system, load gnand.ko, and then load nanddrv.ko.

Device Drivers ->

SCSI device support ->

[\*] SCSI disk support

<M> Nuvoton N9H2x/N329xx/W55FA93 NAND Flash support

-> Select NAND (On board NAND) ->

### **SPI device support**

N9H2x/N329 series Linux BSP supports two CS(Chip Select) of SPI0. When enable one CS, users can select it as spidev or MTD flash. If users want to use the standard spidev interface in user space, they can enable the " User mode SPI device driver support" option. If users want to implement their own driver for their SPI device, they can disable this option.

Device Drivers ->

SPI support ->

--- SPI support

-\*- Utilities for Bitbanging SPI masters

<\*> Nuvoton N9H2x/N329xx/W55FA93 series SPI

SPI CS0 pin select (enable CS0 pin) ->

SPI CS1 pin select (disable CS1 pin) ->

SPI CS0 device select (device for spidev) ->

<\*> User mode SPI device driver support

### **SPI flash support (using MTD support)**

By default, it supports only one MTD partition for SPI flash. Users can modify the partition size and offset. If users want to add new partition tables, they could edit arch/arm/mach-w55fa9x/dev.c to change MTD partition table.

Device Drivers ->

SPI support ->

--- SPI support

-\*- Utilities for Bitbanging SPI masters

<\*> Nuvoton N9H2x/N329xx/W55FA93 series SPI

SPI CS0 pin select (enable CS0 pin) ->

SPI CS1 pin select (disable CS1 pin) ->

SPI CS0 device select (device for MTD flash) ->

<\*> Memory Technology Device (MTD) support ->

--- Memory Technology Device (MTD) support

```
[*] MTD partitioning support
-*- Common interface to block layer for MTD 'translation layers'
<*> Caching block device access to MTD devices
Self-contained MTD device drivers ->
    <*> Support most SPI Flash chips (AT26DF, M25P, W25X, ...)
    (0x00080000) SPI flash partition size
    (0x00380000) SPI flash partition offset
[*] Use FAST_READ OPCode allowing SPI CLK <= 50MHz
```

### **Video/OSD/PWM support**

In general the loader already initials VPOST registers to show Logo, Linux VPOST does not need to initial it again.

If you do not assign Frame buffer address, it may encounter a flicker issue in booting. Due to the buffer address is changed from non-OS to Linux. The Frame buffer address must be the same to loader's setting, and the size is the sum of online Frame and OSD buffers.

"Online + offline buffer" option is used to eliminate the tearing of large screen. And you can enable "H/W PWM" option to control the backlight.

```
Device Drivers ->
    Graphics support ->
        <*> Support for frame buffer devices
            <*> N9H2x/N329xx/W55FA93 LCD framebuffer support
                Select VPOST init setting (vpost init by boot code) ->
                [*] Assign frame buffer starting address and size
                (0x500000) Frame buffer starting address
                (0x4B000) Frame buffer size
                [ ] H/W PWM for backlight control
                    Two frame buffer support (offline + online)? (Two buffer support) ->
                [*] OSD function support
                    Two OSD buffer support (offline + online)? (One OSD buffer support) ->
                [ ] Ping Pong buffer support
                    Select TV/LCM output device (GIANTPLUS GPM1006D0 320X240 LCD support)
->
    Console display driver support ->
        [*] Framebuffer Console support
        [*] Bootup logo ->
            --- Bootup logo
            [*] Standard 224-color Linux logo
```

### **Audio playback/record support**

User can select SPU or I2S for audio playback, and select ADC or I2S for audio recording.

```

Device Drivers ->
  <*> Sound card support ->
    --- Sound card support
    [*] Preclaim OSS device numbers
    <*> Advanced Linux Sound Architecture ->
      --- Advanced Linux Sound Architecture
      <*> OSS Mixer API
      <*> OSS PCM (digital audio) API
      [*] OSS PCM (digital audio) API – Include plugin system
      <*> ALSA for SoC audio support ->
        --- ALSA for SoC audio support
        <*> N9H2x/N329xx/W55FA93 Audio Play Device
          Play device select (N9H2x/N329xx/W55FA93 SPU Support for Playing) ->
          SPU output select (N9H2x/N329xx/W55FA93 SPU without I2S signal output) ->
        <*> Headset Detect and Speaker Control
          Headset detect Enable or Disabled ? (Headset detect Enabled) ->
          Pins select (GPD3 for headset detect and GPD4 for speaker control) ->
          State select for GPD3 (HEADSET plug-in -> GPD3=Low) ->
        <*> SoC Audio for N9H2x/N329xx/W55FA93 series
        <*> N9H2x/N329xx/W55FA93 ADC recording for demo board
  
```

### **AAC support (for N9H26/N3292x only)**

```

Device Drivers ->
  Character devices ->
    [*] Nuvoton N9H2x/N329xx/W55FA92 AAC Device Driver
  
```

### **Video Capture support**

Please select the correct sensor and pins according to the demo board. If you disable “ Specified the packet dimension” option”, it will use the resolution of panel automatically. Please notice only N9H26/N3292x supports the video capture port 2.

```

Device Drivers ->
  <*> Multimedia support ->
    <*> Video For Linux
    [*] Enable Video For Linux API 1 (DEPRECATED)
    [*] Video capture adapters ->
      --- Video capture adapters
      [*] N9H2x/N329xx/W55FA92 video capture port 1 support
        (0x650000) VideoIn DEV1 buffer size
  
```

```

<*>    Support more one planar buffer
< >    Support share sensor to capture port 2
< >    Support motion detection function
        Encode format (Support planar yuv422/yuv420)  ->
<*>    Specified the packet dimension
(320)    Packet pipe width
(240)    Packet pipe height
        Video Sensor support (NT99141 sensor support)  ->
        Video port connection (Data/Control→GPB port)  ->
        Flicker Setting (Flicker setting is 60Hz)  ->
[ ]      N9H2x/N329xx/W55FA92 video capture port 2 support

```

### **JPEG codec support**

Please assign the buffer size of JPEG codec according your requirement.

Device Drivers ->

```

<*> Multimedia support  ->
    HW jpeg codec  ->
        <*> N9H2x/N329xx/W55FA93 HW jpeg codec support
        (0xC8000) Max Raw data size – w*h*byteperpixel(hex) + Max Bistream Size

```

### **H264 decoder support (for N3291x only)**

Please assign the buffer size of H264 decoder according your requirement.

Device Drivers ->

```

Character devices  ->
    <*> W55FA95 Video Decoder support
        Decoded Max Frame Size (1280 x 720(720P))  ->
        (0)
        ax Reference Frame Num (min.=1 and max.=16)
        (0x5C1000) Frame buffer size

```

M

### **H264 codec support (for N9H26/N3292x only)**

You can select enable decoder, encoder or both. And please assign the buffer size of H264 codec according your requirement.

Device Drivers ->

```

[*] Misc devices  ->
    --- Misc devices
    [*] N9H2x/N329xx/W55FA92 H.264 Codec support
        Enabled Codec Driver (H.264 Decoder / Encoder Both)  ->
        Max Frame Size (1280 x 720(720P))  ->

```

- (2) Max Encoder Instance (min.=1 and max.=2)
- (1) Max Decode Reference Frame Num (min.=1 and max.=16)
- (0xB72000) Frame buffer size

### **VPE support (for N3291x and N9H26/N3292x only)**

Device Drivers ->

Character devices ->

[\*]N9H2x/N329xx/W55FA92 Video Post-processing Engine support

### **BLT support (for N9H20/N3290x and N9H26/N3292x only)**

Device Drivers ->

Character devices ->

<\*> Nuvoton N9H2x/N329xx/W55FA92 BLT Device Driver

### **OVG support (for N3291x only)**

OpenVG supports both MMU and non-MMU mode, the default setting is MMU mode.

Device Drivers ->

Character devices ->

<\*> Nuvoton W55FA95 OpenVG Device Driver

[ ] Support OpenVG non-MMU mode

### **Keypad (GPIO) support (for N9H20/N3290x and N3291x only)**

This configuration is board dependent., follow option may only workable on demo board. In generally, users need to develop the keypad driver for different board.

Device Drivers ->

Input device support ->

<\*> Event interface

[\*] Keyboards ->

[\*] Nuvoton N9H2x/N329xx/W55FA93 Keypad Support

Keypad type (2 x 3 event interface) ->

### **Keypad (KPI) support (for N3291x only)**

Device Drivers ->

Input device support ->

<\*> Event interface

[\*] Keyboards ->

[\*] Nuvoton W55FA95 Keypad Support

Keypad type (M(ROW) x N(COL) using KPI) ->

(11) Keypad row number (min.=2 and max.=16)

- (1) Keypad column number (min.=1 and max.=4)
- Keypad value (key value normal) ->

**Touch Screen/Voltage detection support (for N9H20/N3290x and N3291x only)**

Device Drivers ->

Input device support ->

<\*> Event interface

[\*] Touchscreens ->

<\*> N9H2x/N329xx/W55FA93 touchscreens

[\*] Detect ambient light and battery status

**Keypad/Touch Screen/Voltage detection support (for N9H26/N3292x only)**

Device Drivers ->

Input device support ->

<\*> Event interface

[\*] Miscellaneous devices ->

[\*] N9H2x/N329xx/W55FA92 Input ADC Function

ADC Function Select ->

[\*] Touch panel detection

[\*] Battery detection

[\*] Keypad detection

BATTERY detection channel selection (Battery detection from channel 1) ->

Keypad detection channel selection (Keypad detection from channel 2) ->

**USB host support (EHCI driver is for N9H26/N3292x only)**

Here list the configuration to enable OHCI/EHCI for USBH interface to support mass storage device. To support mass storage device. Except the configuration shown below, SCSI disk support also needs to be enabled.

Device Drivers ->

[\*] USB support ->

--- USB support

<\*> Support for Host-side USB

[\*] USB device class-devices (DEPRECATED)

<\*> EHCI HCD (USB 2.0) support

[\*] N9H2x/N329xx/W55FA92 EHCI support

<\*> OHCI HCD support

[\*] Nuvoton N9H2x/N329xx/W55FA92 OHCI support

[ ] Nuvoton N9H2x/N329xx/W55FA92 OHCI Like support



```
<*> USB Mass Storage support
SCSI device support ->
<*> SCSI device support
<*> SCSI disk support
[*] Probe all LUNs on each SCSI device
```

### **USB device support**

Below list the configuration for USB mass storage support. To install the kernel module, use command: **insmod g\_file\_storage.ko file=<device1>,<device2> stall=0 removable=1**

```
Device Drivers ->
[*] USB support ->
--- USB support
<*> USB Gadget Support ->
--- USB Gadget Support
      USB Peripheral Controller (N9H2x/N329xx/W55FA92 USB Device Controller) ->
<M> USB Gadget Drivers
<M> File-backed Storage Gadget
```

### **Watchdog timer support**

```
Device Drivers ->
[*] Watchdog Timer Support ->
--- Watchdog Timer Support
<*> Nuvoton N9H2x/N329xx/W55FA93
```

### **RTC support**

```
Device Drivers ->
<*> Real Time Clock ->
--- Real Time Clock
[*] Set system time from RTC on startup and resume
(rtc0) RTC used to set the system time
[*] /sys/class/rtc/rtcN (sysfs)
[*] /proc/driver/rtc (procfs for rtc0)
[*] /dev/rtcN (character devices)
<*> Nuvoton N9H2x/N329xx/W55FA93
```

### **I2C support**

Users could choose to use the native I<sup>2</sup>C interface, or use GPIO pins to simulate SDA and SCL.

```
Device Drivers ->
<*> I2C support ->
```

```

--- I2C support
I2C Hardware Bus support  ->
    <*> N9H2x/N329xx/W55FA93 I2C Driver
    < > Nuvoton N9H2x/N329xx/W55FA93 I2C using GPIO pins
    
```

### **GPIO support**

Users could enable below options to support control GPIO interface in user space.

```

Device Drivers  ->
    *- GPIO Support  ->
        --- GPIO Support
        [*] /sys/class/gpio/... (sysfs interface)
        <*> N9H2x/N329xx/W55FA93 GPIO support
    
```

### **AES support (for N3291x and N9H20/N3292x only)**

```

Device Drivers  ->
    [*] Block devices  ->
        --- Block devices
        <*> Loopback device support
        <*> Cryptoloop Support
    *- Cryptographic API  ->
        --- Cryptographic API
        [*] Hardware crypto devices  ->
            <*> W55FA95 AES H/W accelerator
        <*> CryptoDev for Linux
    
```

### **EMAC support (for N9H26/N3292x only)**

Please select correct PHY chip and N9H26/N3292x MAC

```

Device Drivers  ->
    [*] Network device support  ->
        --- Network device support
        [*] Ethernet (10 or 100Mbit)  ->
            --- Ethernet (10 or 100Mbit)
            [*] Nuvoton N9H2x/N329xx/w55fa92 Ethernet support
                Ethernet port select (N9H2x/N329xx/W55FA92 Ethernet 1st Port)  ->
            [*] RTL8201 PHY
            [*] RTL8201 PHY
    
```

## 6.3. File system support

### ***FAT-based file systems(MS-DOS, VFAT) support***

```
File systems ->
  DOS/FAT/NT Filesystems ->
    <*> MSDOS fs support
    <*> VFAT (Windows-95) fs support
    (437) Default codepage for FAT
    (iso8859-1) Default iocharset for FAT
  Partition Types ->
    [*] Advanced partition selection
    [*] PC BIOS (MSDOS partition tables) support
  -* Native language support ->
    --- Native language support
    (cp437) Default NLS Option
    <*> Codepage 437 (United States, Canada)
    <*> NLS ISO 8859-1 (Latin 1; Western European Languages)
```

### ***ROM file system support***

```
File systems ->
  [*] Miscellaneous filesystems ->
    --- Miscellaneous filesystems
    <*> ROM file system support
    RomFS backing stores (Block device-backed ROM file system support) ->
```

### ***JFFS2 file system support***

```
File systems ->
  [*] Miscellaneous filesystems ->
    --- Miscellaneous filesystems
    <*> Journalling Flash File System v2 (JFFS2) support
    (0) JFFS2 debugging verbosity (0 = quiet, 2 = noisy)
    [*] JFFS2 write-buffering support
```

### ***/proc and sysfs file system support***

```
File systems ->
  Pseudo filesystems ->
```

```
[*] /proc file system support
[*]   Sysctl support (/proc/sys)
[*]   Enable /proc page monitoring
[*] sysfs file system support
```

## 6.4. Network support

### **TCP/IP support**

To enable TCP/IP network, please follow the setting below:

```
[*] Networking support ->
    --- Networking support
    Networking options ->
        <*> Packet socket
        <*> Unix domain sockets
        [*] TCP/IP networking
        [*]   IP: multicasting
```

### **Wireless support**

To support Wireless feature, please select “Nuvoton external WiFi driver support “ to enable WIRELESS\_EXT and WEXT\_PRIV options at first. The NL80211 is necessary to some WiFi AP mode drivers.

```
Device Drivers ->
    [*] Network device support ->
        --- Network device support
        [*]   Wireless LAN ->
            --- Wireless LAN
            <*>   Nuvoton external WiFi driver support
[*] Networking support ->
    --- Networking support
    -* Wireless ->
        --- Wireless
        <*>   cfg80211 – wireless configuration API
        [*]   nl80211 testmode command
        [*]   enable powersave by default
        [*]   cfg80211 wireless extensions compatibility
        [*]   Wireless extensions sysfs files
```

## 7. Hardware and Driver Source Location

Follow table shows the N9H2x/N329 series hardware IP and the location of the driver source. The device name is the interface that a application connects to a driver. If you want to support a new feature of a driver, you can modify the driver source to meet your requirement.

Hardware	Device Name	Driver Location
AAC	/dev/aac	drivers/char/aac/ (N9H26/N3292x only)
ADC Keypad	/dev/input/event[N]	drivers/input/misc/w55fa92_adc/keypad.c (N9H2x/N329xx only)
ADC Touch	/dev/input/event[N]	drivers/input/touchscreen/w55fa93_ts.c (for N3290x and N3291x) drivers/input/misc/w55fa92_adc/touch.c (for N9H26/N3292x)
ADC Voltage	/sys/devices/platform/w55fa9x-adc/ad	drivers/input/touchscreen/w55fa93_ts.c (for N9H20/N3290x and N3291x) drivers/input/misc/w55fa92_adc/battery.c (for N9H26/N3292x)
AES	None	drivers/crypto/w55fa9x-aes/ (N3291x and N9H26/N3292x only)
BLT	/dev/blt	drivers/char/blt/ (N9H20/N3290x and N9H26/N3292x only)
CPU Clock	sys/devices/platform/w55fa9x-clk/clock	arch/arm/mach-w55fa9x/cpu.c
EDMA	None	arch/arm/mach-w55fa9x/edma.c, DrvEDMA.c
EMAC	eth0	drivers/net/arm/w55fa92_*.c (N9H2x/N329xx only)
GPIO/KPI Keypad	/dev/input/event[N]	drivers/input/keypad/w55fa9x_keypad_*.c
H264 Decoder	/dev/vde	drivers/char/vde/ (N3291x only)
H264 Codec	/dev/w55fa92_264dec, /dev/w55fa92_264enc	drivers/misc/codec264/ (N9H2x/N329xx only)

I <sup>2</sup> C	/dev/i2c0	drivers/i2c/buses/i2c*-w55fa9x.c
JPEG	/dev/video[N]	drivers/media/jpeg/w55fa9x/
NAND	/dev/sd[a-d][N]	drivers/scsi/ uvoton_nand/
OpenVG	/dev/gshark	drivers/char/ovg/ (N3291x only)
RTC	/dev/rtc0	drivers/rtc/rtc-w55fa9x.c
SD	/dev/mmcblk[N]	drivers/mmc/core/sd.c, drivers/mmc/host/w55fa9x_sd.c
SDIO	/dev/mmcblk[N]	drivers/mmc/host/w55fa92_sdio.c (N9H2x/N329xx only)
SPI	/dev/spi[N]	drivers/spi/spi_w55fa9x.c
SPU ADC Audio I2S	/dev/dsp[N], /dev/mixer[N] (for OSS)  /dev/snd/ (for ALSA)	sound/soc/w55fa9x (for SPU/I2S audio playback, and I2S audio recording)  sound/soc/w55fa9xadc (for ADC audio recording)
UART	/dev/ttyS[N]	drivers/serial/w55fa9x_serial.c
USBD	None	drivers/usb/gadget/w55fa9x_udc.c
USBH 1.1	None	drivers/usb/host/ohci-w55fa9x.c  drivers/usb/host/ohci-like-w55fa92.c (N9H26/N3292x only)
USBH 2.0	None	drivers/usb/host/ehci-w55fa92.c (N9H2x/N329xx only)
Video-In	/dev/video[N]	drivers/media/video/w55fa9x_*/
VPE	/dev/vpe	drivers/char/vpe/ (N3291x and N9H26/ N3292x only)
VPOST PWM	/dev/fb0	drivers/video/w55fa9x_*.c
Watchdog	/dev/watchdog	drivers/char/watchdog/w55fa9x_wdt.c

## 8. Sample Applications

There are some sample applications in the applications/ directory. Content of each directory listed in the following table.

Directory	Description
aacdec	AAC decode sample application (N9H26/N3292x only)
aacenc	AAC encode sample application (N9H26/N3292x only)
aes	AES sample application (N3291x and N9H26/N3292x only)
alsa_demo	Audio sample code of ALSA
audio_oss2alsa	Audio playback and record sample application. Which uses OSS APIs to connect ALSA drivers
backlight	PWM backlight control sample application
blt / bitblt	BLT sample application (N9H20/N3290x and N9H26/N3292x only)
busybox-1.15.2	Busybox source code. The build sequence listed below: <ol style="list-style-type: none"> <li>1. Copy the default configuration to file ".config"</li> <li>2. Type "make menuconfig" command to enter busybox configuration menu</li> <li>3. Modify the options you want to add or delete</li> <li>4. Type "make" command to build busybox</li> <li>5. Type "make install" command to install busybox to rootfs</li> </ol>
devMem	Device memory sample application
emWin	Demo code of emWin GUI
gen_lib	Generate a ROM image of libraries for dynamic linking applications
gpio	User space GPIO control sample
h264codec	H264 encode/decode sample application (N9H26/N3292x only)
hid_gadget_test	USB gadget HID sample
hid_gadget_vendor	USB gadget vendor command sample

jpeg_codec	JPEG encode/decode sample application
keypad	Keypad sample application
lcm	Frame buffer sample application
libjpeg_HW	IJG Jpeg codec library and sample support H/W JPEG
lvd	Low voltage detection sample application
mtdnand	MTD NAND solution
osd	OSD sample application
ovg	OpenVG sample application (N3291x only)
powerManager	Power management sample application
printer_gadget	USB gadget printer sample
rot	ROT sample application (N9H26/N3292x only)
rtc	RTC sample application
serial_loop	Serial loopback transfer test sample
serial_transfer	Serial transfer sample of client and server
spi	SPI flash raw data read/write test
spidev	SPI testing utility using spidev driver
touchscreen	Touch screen sample application
tslib-1.1 / tslib-master	Tslib source code
usb_plug	USB device plug-in/out control for mass storage
uvc_gadget_demo	USB gadget UVC sample
vde	H264 decoder sample application (N3291x only)
videoin_HW	Video-in capture sample application using H/W preview mode
vpe	VPE sample application (N3291x and N9H26/N3292x only)
watchdog	Watch dog timer sample application
webcam_demo	USB host support webcam sample



## 8.1. Cross Compilation Makefile

Sometimes a project requires porting an application to ARM platform. If the application's Makefile doesn't support cross compilation options, the modification of Makefile is necessary. The Makefile used for cross compiling could be alike with the original one, only part of it needs to be modified.

- The prefix of tool chain must be set. For example, the original Makefile use gcc for compiling, the new Makefile use arm-linux-gcc for cross compiling.
- The path of library and include files need to be set. The cross compiler doesn't use the glibc or other library using in x86 system.
- The `–static` option must be given for static linking.
- If the `–static` option is not found, it will use dynamic linking. And you need to provide the ROM image of libraries which your applications need.

Here is a sample Makefile for static link.

```
.SUFFIXES : .x .o .c .s

ROOT = /usr/local/arm_linux_4.2/
LIB = $(ROOT)/lib/gcc/arm-linux/4.2.1
LIB1 = $(ROOT)/arm-linux/lib
INCSYS:= $(ROOT)/arm-linux/sys-include
INC := $(ROOT)/arm-linux/include

CC=arm-linux-gcc -O2 -I$(INC) -I$(INCSYS) -static
WEC_LDFLAGS=-L$(LIB) -L$(LIB1)
STRIP=arm-linux-strip

TARGET = hello

SRCS := hello.c

LIBS= -lc -lgcc -lc
all:
    $(CC) $(WEC_LDFLAGS) $(SRCS) -o $(TARGET) $(LIBS)
    $(STRIP) $(TARGET)

clean:
    rm -f *.o
```

```
rm -f $(TARGET)
```

```
rm -f *.gdb
```

## 9. Advance Control in Linux BSP

There are some advance system control items in Linux BSP, and below are the detail introductions.

---

### 9.1. How to enable core dump feature?

Generate and parse core dump is a well-known skill for Linux application debugging. To enable core dump, you must do below procedures.

1. In kernel configuration, enable "General setup -> Configure standard kernel features (for small systems) -> Enable ELF core dumps". The default N9H2x/N329x configurations already enable this option.
2. In busybox configuration, enable "Init Utilities -> Support dumping core for child processes". The default N9H2x/N329x configurations already enable this option.
3. In BSP/rootfs-2.6.35 folder, rename file ".\_\_init\_enable\_core\_\_" to be ".init\_enable\_core".
4. Rebuild the kernel image.

This section only introduces how to enable core dump feature in our Linux BSP. About how to use it, you can find some information from internet.

---

### 9.2. How to enter to suspend mode and resume?

#### **N9H2X/N3290x**

Run below command can enter to suspend mode.

```
Echo pd > /sys/devices/platform/w55fa93-clk/clock
```

The default wakeup sources of N9H26/N3290x Linux are GPIO keypad, touch screen and RTC.

#### **N3291x**

Run below command can enter to suspend mode.

```
Echo pd > /sys/devices/platform/w55fa95-clk/clock
```

The default wakeup sources of N3291x Linux are GPIO keypad, KPI and RTC.

#### **N9H26/N3292x**

Run below command can enter to suspend mode.

```
Echo pd > /sys/devices/platform/w55fa92-clk/clock
```

The default wakeup sources of N3292x Linux are GPIO keypad, ADC keypad, EMAC and RTC.

---

## 9.3. How to dynamic change CPU and PLL clocks?

### **N9H20/N3290x**

Run below command to adjust the CPU and UPLL clock.

Case 1: Adjust only CPU clock.

*Echo CPU\_CLOCK > /sys/devices/platform/w55fa93-clk/clock*

In which CPU\_CLOCK means N[MHz] clocks, such as 240, 192 or 144. Please notice the CPU clock must be a calculable value from UPLL clock.

Case 2: Adjust both of CPU and UPLL clocks.

*Echo CPU\_CLOCK:UPLL\_CLOCK > /sys/devices/platform/w55fa93-clk/clock*

In which CPU\_CLOCK and UPLL\_CLOCK mean N[MHz] clocks, such as 240:240, 96:192 or 36:144. Please notice the UPLL clock must be a calculable value from external clock, and the CPU clock must be a calculable value from UPLL clock.

### **N3291x**

In above commands, use w55fa95 instead of w55fa93.

### **N9H26/N3292x**

In above commands, use w55fa92 instead of w55fa93.

# 10. Revision History

Version	Date	Description
V1.0	Aug, 8. 2013	<ul style="list-style-type: none"> <li>• Created</li> </ul>
V1.1	Oct, 17. 2013	<ul style="list-style-type: none"> <li>• Update BSP information of N3292x</li> </ul>
V1.2	Dec, 20. 2013	<ul style="list-style-type: none"> <li>• BSP supports N3291x and update some information</li> </ul>
V1.3	Jan, 28, 2014	<ul style="list-style-type: none"> <li>• Add board information of N32925 and N3291x</li> </ul>
V1.4	Jul, 24, 2014	<ul style="list-style-type: none"> <li>• Add a section for advance system control</li> </ul>
V1.5	Sep, 20, 2018	<ul style="list-style-type: none"> <li>• BSP supports N9H2x and update some information.</li> </ul>
V1.6	Feb, 11, 2020	<ul style="list-style-type: none"> <li>• Add missing packages in CentOS / Ubuntu</li> </ul>

### Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments, aircraft or spacecraft instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for any other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications whereby failure could result or lead to personal injury, death or severe property or environmental damage.

Nuvoton customers using or selling these products for such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from their improper use or sales.